BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

(TRANSLATION)

Japanese Patent Application No. 08-241335
Patent Date : September 17, 1996

Application No. : 7-280952

Filing Date : October 27, 1995

Applicant : CANON INC

Inventor (s) : KENISU EMU HANTAA, et al.

Title of the Invention :
METHOD AND SYSTEM FOR VAGUE CHARACTER STRING RETRIEVAL USING FUZZY INDETERMINATIVE FINITE AUTOMATION

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-241335
(43)Date of publication of application : 17.09.1996

(51)Int.Cl. G06F 17/30

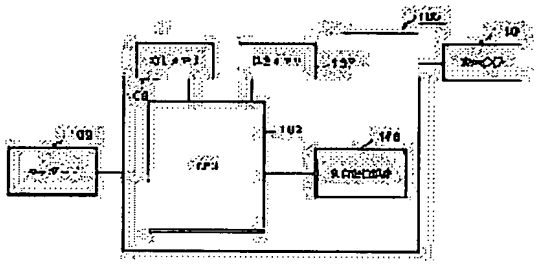
(21)Application number : 07-280952
(22)Date of filing : 27.10.1995

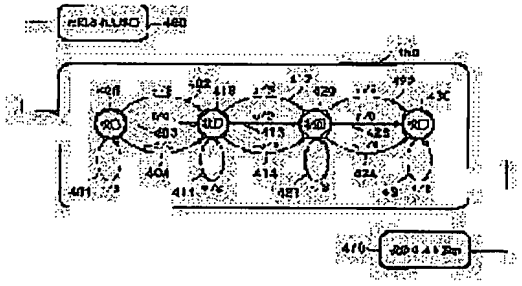
(71)Applicant : CANON INC
(72)Inventor : KENISU EMU HANTAA
MAIKERU JII ROBAATSU
HARII TEI GAARANDO

(30)Priority
Priority number : 94 330968 Priority date : 28.10.1994 Priority country : US

(54) METHOD AND SYSTEM FOR VAGUE CHARACTER STRING RETRIEVAL USING FUZZY INDETERMINATIVE FINITE AUTOMATION

(57)Abstract:
PURPOSE: To provide the method and system which selectively retrieve information included in a stored document set by using the 'fuzzy' indeterminative finite automaton by metrics method and the non-literal retrieval method.
CONSTITUTION: The system is provided which has a data input part 104 for transmitting a user-defined text string inquiry to a processor 102, the indeterminative finite automaton 450 constituted so as to correspond to the text string inquiry, and a display part 470 providing a user with a classification list of stored words having discrepant values below a threshold value, and the stored words 460 are applied to the automaton 450, which generates discrepant values regarding the thus stored words respectively.





LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平8－241335

(43)公開日　平成8年（1996）9月17日

(51)Int.Cl.[6]	識別記号	庁内整理番号	FI		技術表示箇所
G06F 17/30		9194－5L	G06F 15/403	350	C
		9194－5L	15/40	380	C

審査請求　未請求　請求項の数27　OL（全 26 頁）

(21)出願番号　特願平7－280952

(22)出願日　平成7年（1995）10月27日

(31)優先権主張番号　08／330968

(32)優先日　1994年10月28日

(33)優先権主張国　米国（US）

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号

(72)発明者　ケニス・エム・ハンター
アメリカ合衆国　カリフォルニア州　94414、　サンフランシスコ、　サターン　ストリート　エー151

(72)発明者　マイケル・ジー・ロバーツ
アメリカ合衆国　カリフォルニア州　94041、　マウンテンビュー、#3201　ハイスクール　ウェイ　950

(74)代理人　弁理士　大塚　康徳　（外１名）

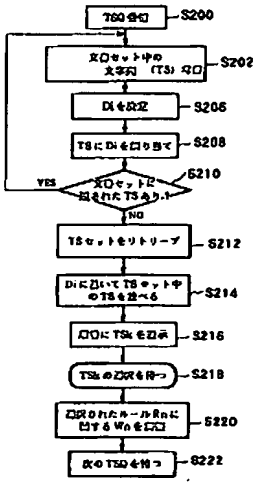
最終頁に続く

(54)【発明の名称】ファジー非決定性有限オートマトンを使用したあいまいな文字列検索方法及びシステム

(57)【要約】

【課題】　メトリクス法による、「ファジー」な非決定性有限オートマトンを用いて、非リテラル検索方法を使用することによって、記憶された文書セットに含まれる情報を選択的にリトリーブするための方法及びシステムを提供することを目的とする。

【解決手段】　ユーザー定義によるテキストストリング照会を処理装置（１０２）へ送信するためのデータ入力部（１０４）と、このテキストストリング照会に対応するように構成されている非決定性有限オートマトン（４５０）としきい値以下の不一致値を持つ記憶された単語の分類リストをユーザーに提供する表示部（４７０）とを有するシステムを提供し、記憶された単語（４６０）はオートマトン（４５０）に適用され、オートマトン（４５０）はこのように記憶された単語それぞれに関する不一致値を生成する。



1

【特許請求の範囲】

【請求項1】　コンピューター処理部によってアクセスすることのできるデータ記憶媒体に記憶されている文書セットに含まれている複数の文字列を含む情報を選択的にリトリーブするための、コンピューターによって実施される方法であって、

ユーザーによって定義された文字列照会を前記処理部へ送信する工程と、

前記文字列照会に対応するファジー非決定性有限オートマトンを構成する工程と、

前記記憶された文字列を前記オートマトンに適用することによって、前記記憶された文書セット中の前記文字列のそれぞれに関連する累計不一致メトリクスを生成する工程と、

サブセット中の前記複数の文字列の各々に関連する累計不一致メトリクスに応答して、前記複数の記憶された文字列の前記サブセットを表示する工程とを有することを特徴とする方法。

【請求項2】　前記オートマトンを構成する工程は、前記文字列照会と前記文字列間の予め決められた差異を求めるために、遷移ペナルティー値を前記記憶された文書セット中の文字列に関連する不一致度メトリクスの累計に加算する装置を提供することを含むことを特徴とする請求項1に記載の方法。

【請求項3】　予め決められた差異は不足文字に関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項4】　予め決められた差異は余分な文字に関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項5】　予め決められた差異は曖昧な文字に関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項6】　予め決められた差異は音声に基づく文字の置き換えに関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項7】　予め決められた差異は文法による置き換えに関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項8】　予め決められた差異は大文字への変更に関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項9】　予め決められた差異は綴り間違えに関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項10】　予め決められた差異は近隣の文字の順序交換に関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項11】　予め決められた差異は漢字間違えに関する差異を含むことを特徴とする請求項2に記載の方法。

【請求項12】　前記文字列の一つの文字列のための前記オートマトンの処理を、前記一つの文字列に関連する累計不一致メトリクスが最大しきい値に達した場合、このメトリクスに応じて中断する工程を更に有することを特徴とする請求項2に記載の方法。

2

【請求項13】　文書に対応するサブセット中のストリングそれぞれに関連する不一致度メトリクスを合計することによって、記憶された文書セット中の文書の総計不一致度メトリクスを確立する工程を更に有することを特徴とする請求項2に記載の方法。

【請求項14】　コンピューター処理部によってアクセスすることのできるデータ記憶媒体に記憶されている文書セットに含まれている複数の文字列を含む情報を選択的にリトリーブするためのコンピューターシステムであって、

ユーザーによって定義された文字列照会を前記処理部へ提供するデータ入力手段と、

前記文字列照会に対応し、前記記憶された文字列を入力として受け入れ、それに応じて前記記憶された文字列それぞれに関連する不一致メトリクスを生成するファジー非決定性有限オートマトンと、

サブセット中の前記複数の文字列の各々に関連する累計不一致メトリクスに応答して、前記複数の記憶された文字列の前記サブセットを表示する表示手段とを有することを特徴とするコンピューターシステム。

【請求項15】　オートマトンは、前記文字列照会と前記文字列間の予め決められた差異を求めるために、ペナルティー値を記憶された文書セット中の文字列に関連する不一致度メトリクスに加算するように構成されていることを特徴とする請求項14に記載のコンピューターシステム。

【請求項16】　予め決められた差異は不足文字に関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項17】　予め決められた差異は余分な文字に関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項18】　予め決められた差異は認識不能文字に関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項19】　予め決められた差異は音声に基づく文字の置き換えに関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項20】　予め決められた差異は文法による置き換えに関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項21】　予め決められた差異は大文字への変更に関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項22】　予め決められた差異は綴り間違えに関する差異を含むことを特徴とする請求項15に記載のコンピューターシステム。

【請求項23】　予め決められた差異は近隣の文字の順

序交換に関する差異を含むことを特徴とする請求項１５に記載のコンピューターシステム。

【請求項２４】　予め決められた差異は漢字間違えに関する差異を含むことを特徴とする請求項１５に記載のコンピューターシステム。

【請求項２５】　前記オートマトンは、前記文字列の内一つの文字列に関連する累計不一致メトリクスが最大しきい値に達した場合、このメトリクスに応じて前記一つの文字列に関する処理を中断することを特徴とする請求項１５に記載のコンピューターシステム。

【請求項２６】　前記オートマトンは、文書に対応するサブセット中のストリングそれぞれに関連する不一致度メトリクスを合計することによって、前記記憶された文書セット中の文書の総計不一致度メトリクスを確立することを特徴とする請求項１５に記載のコンピューターシステム。

【請求項２７】　コンピューター処理部によってアクセス可能に記憶されている文書セットに含まれている複数の文字列を含む情報を、前記コンピューター処理部によって選択的にリトリーブされるメモリ媒体であって、
ユーザーによって定義された文字列照会を前記処理部へ送信する工程の手順コードと、
前記文字列照会に対応するファジー非決定性有限オートマトンを構成するの工程手順コードと、
前記記憶された文字列を前記オートマトンに適用することによって、前記記憶された文書セット中の前記文字列のそれぞれに関連する累計不一致メトリクスを生成する工程の工程手順コードと、
サブセット中の前記文字列各々に関連する累計不一致メトリクスに応答して、前記複数の記憶された文字列の前記サブセットを表示する工程の工程手順コードを備えることを特徴とするメモリ媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複合文書の操作において使用されるタイプの文書保存及びリトリーブ（retrieve）システム関し、特に、メトリクスを基本とする、つまり「ファジー（曖昧）」な非決定性有限オートマトンを用いた非リテラル検索方法を用いて、保存された文書から選択的に情報を検索及びリトリーブするための方法及びシステムに関する。

【０００２】

【従来の技術】文書の電子保存により、病院や大学、政府機関等で取り扱われる文書等の、膨大な文書の取り扱いが容易にできるようになった。通常、このような文書は、タイプによる直接入力や、電子メールの受信や、走査入力を含む様々な手段により巨大な記憶システムに入力される。走査システムはしばしば、走査した画像のテキスト部分を電子データへ変換する光学式文字読み取り装置（OCR）を有する。保存された文書はこのように、

画像と、テキストと、キーワードのような注釈とが混在することもあり、様々な電子形態で記憶することができる。記憶された文書セットから選択的に情報をリトリーブ（retreive）することは、検索しなければならない情報量が増加するに従い、深刻な問題となっている。

【０００３】従来のアーカイブ及びリトリーブシステムは、様々な検索技術を支えている。これらの技術には、自動又はユーザー定義によるインデックス付与や、キーワード注釈、自動キーワード抽出、フルテキスト検索、テキスト中の部分的又は全ての単語又は句に対する前処理インデックス付与や、リテラル及び非リテラル検索等がある。

【０００４】典型的な従来のシステムは、文書が記憶部に入力される時にそれぞれの文書にインデックスを割り当てる。このインデックスはそれぞれの文書に関してシステムが生成したコードでも、ユーザーが定義したコードでもよい。そして、このコードは文書と共に記憶される。文書をリトリーブするためには、ユーザーは目的とする文書に関連付けられた適切なコードを入力しなければならない。キーワードを同様の方法で使用するシステムもある。キーワードを認識して文書に割り当てる方法としては、ユーザーによるキーボードからの直接入力や、ユーザーによる文書テキストからの対話的選択や、文書テキストの検索による自動抽出等、様々な方法がある。キーワードが文書に割り当てられた後は、ユーザーは文書をリトリーブするためにキーワードを使用することができる。このようなシステムにおける問題点は、ユーザーは文書全体をリトリーブしなければならず、また、目的の文書に関連付けられたインデックスや、コード、又はキーワードを知っていなければならないことである。

【０００５】フルテキスト検索システムでは、システム中へ検索語を入力することによって設定された、文書中の選択された情報にユーザーがアクセスすることができる。そしてこのシステムは入力された検索語に完全に一致するものを検出するために、設定された文書全体を読み通す。この作業では、文書テキスト中のストリングのある特定の例示位置が分かるという利点がある。これらの位置は、近接検索等を容易にする。この検索では、検索表現がそのある部分に一致する文書セットの文字列の相対的な位置について、制限を含むことがある。このようなシステムにおける問題点は、それぞれの検索が文書セットテキスト全体を確認する工程を含むことであり、非常に多数の文書セットがある場合に検索スピードを遅くしてしまうことである。

【０００６】予め処理された又はインデックスを付された検索システムは、通常、文書セットテキスト中に存在する単語表を作成する。これらの表の使用により、多量の文書セットを検索する際の効率が向上する。例えば、簡単な例では、先ず最初に表の検索を行い、そして、望

ましい目的の単語が含まれていると表から判断できる文書のみを更に検索する。特定の状況では、動作を最適化するためにこの表をソートしたり、クロスインデックスを付与したりしてもよい。

【０００７】しかし、フルテキスト及びインデックス付与による検索システムでは、検索語と文書セット間で不一致が起こることが時々ある。例えば、検索語を入力するときに、キーボードによる入力ミスや他のエラーにより、ユーザーが間違った又は意図しない検索語を入力することもある。他の例では、原稿のテキストや、OCRや、手動で入力されたキーワードにエラーがある場合がある。完全な一致を要求するリテラル検索システムは、入力された検索語と文書セットのテキスト間のこのような不一致を取り扱うことができず、このような場合、目的の文書をリトリーブできないことがある。

【０００８】非リテラル、つまり「ファジー」な検索システムは不一致を取り扱うことができる。このようなシステムでは、文字列をコンピューターシステムに入力し、保存されたテキストファイル中の文字列から「近似的に」一致するものを検索する動作を含む。例えば、ユーザーが「recieve」（綴りが違っている）の検索をリクエストしたとすると、システムは正しく綴られた単語、「receive」を検出することができる。他の例として、OCRから得られる光学走査された文書のテキストファイルが記憶されている場合、そのOCRシステムはしばしば形状が似ている文字を誤って認識することがあるとする。例えば、文字「O」（アルファベット）を数字の「０」と誤識したり、２つのアルファベットの組み合わせ「rn」を１つのアルファベット「m」と誤認することがある。このような場合、入力文字列に形状が似ているテキストをリトリーブすることが望ましい。

【０００９】公知のファジー検索技術は、検索語に「近似した」単語を含む文書を見つける課題に充分に応用されてはいない。例えば、COMUNICATIONS OF THE ACM 35, 10(１９９２年１０月）の７４ページから８２ページに掲載されたR. Baeza-YatesとG. Gonnet著の「新しいテキスト検索法（A New Approach to Text Searching）」では、不一致文字を含む目的単語と検索語間の一致を見つける技術が示されているが、不足文字や、余分な文字、又は近隣の文字の一部が逆になっている単語をうまく取り扱う技術は示されていない。第２の技術として、COMUNICATIONS OF THE ACM 35, 10(１９９２年１０月）の８３ページから９１ページにS. WuとU. Manber著の「エラーを容認する高速テキスト検索」が掲載されているが、これは、適応するコスト精密調整が望ましい場合に、不一致文字や、不足文字、余分な文字に関連する小整数コストを利用することにより、これらのコストを精密にあわせる能力を厳しく制限することを示している。更にこれらの技術では、近隣の文字の一部が逆になった単語を不足及び余分な文字の混合として捉えているため、近隣の文字の一部が逆になった単語のコストは単に過不足文字のコストの合計として見つけられる。ファジー検索を実施するために、WuとManberの技術によれば、エラーのない一致を最初に検索し、次に１エラーがあるものを検索すると言った動作を充分な一致が検出されるまで続ける。

【００１０】第３の技術は、やはりU. ManberとS. Wuにより開発されたもので、スイス連邦国ベルンで開かれた構造及びシンタックス認識についてのLAPR研究会で１９９０年２月及び１９９２年８月に発行された「テキスト及びハイパーテキストのための任意コストによる近似ストリング一致（Approximate String Matching with Arbitrary Costs for Text and Hypertext）」に示されている、過不足文字を取り扱う技術に関するものである。著者は、「アルゴリズムを１回でも控除すると、代わりのものを取り扱うことができなくなる。つまり、１つの文字を他の文字で置き換えることのコストは、第一の文字を削除し、そして第二の文字を挿入するコストと同じであると見做す。」と記述している。近隣の文字が逆になった単語についても、同様の問題がある。

【００１１】第４の技術は、１９９２年にFujisawaらによって米国特許番号第４，９８５，８６３で開示されており、これによると、完全一致のためのリテラル検索のみを行うために有限決定性オートマトンを使用しているが、OCRが殆ど確実性のない場合にOCR文書テキストの文字の他のアイデンティティー（identity）へと符号化する。これは、不足文字や、余分な文字や、近隣の逆になった文字を取り扱う方法を提示するものではないため、不一致文字の一般的な取り扱いを何ら示していない。

【００１２】これらの技術のそれぞれはある特殊な限られた場合に適するかもしれないが、検索語中のエラーの数及び種類が分からない場合に、検索語に基づいて文字列を見つける一般的な利用においては不便である。この制限は文書セット中に存在する異なる単語の数が膨大になるにつれて、特に深刻な問題となる。コンピューターシステムにおける有限状態オートマトンにおいて、ある特定のパターンにシンボル列が一致しているか否かを決定するためにそのシンボル列を分解する使用方法が知られている。分析されたシンボルは、ASCII文字コード等の有限シンボルセットのメンバーである。オートマトンは初期状態又は初期状態のセットから動作を開始し、入力されるシンボル列を順番に処理する。入力するシンボルそれぞれが処理される度に、オートマトンの過去の状態及び入力するシンボルのアイデンティティーに応じてオートマトンの状態は変化する。もし、入力最終シンボルが処理されてオートマトンが最終状態に達したときは、その入力シンボル列は認識のためにオートマトンを構成したある特定のパターンに一致するシンボルであることが分かる。それ以外は、そのシンボル列は認識のためにオートマトンを構成したある特定のパターンに一致

しないことが分かる。

【0013】オートマトンは決定性型でも非決定性型でもよい。決定型オートマトンは、それぞれの時間ポイントで単一の現状態を持ち、次に検索される特定のシンボルが存在する。最も簡単な場合では、次のシンボルを処理した結果は、オートマトンが継続する現状態の一つへ移行することであり、この移行した状態は現在と同じ状態であるかもしれないが、どの場合も前の状態や入力シンボルによって完全に決定される。この処理は全てのシンボルが処理されるか、最終状態に到達するか、これ以上の有効な入力がないことを示す入力文字が受信されるまで続けられる。

【0014】決定性オートマトンのデザインと、状態の継続と、入力シンボルによっては、実行可能な次の状態が1つ以上存在する場合がある。ある時点で1つの状態だけ最新であるために、実行可能な次の状態がある度にその回数だけオートマトンは複製され、それぞれの複製は状態順列と次のシンボルにより異なる経路を辿る。この状態順列木は非常に多くの論理出力を持つことができるため、非常に非効率的な処理となる。引き返し処理を行ったとしても、上記処理は根本的に非効率的である。状態のある特定の経路が最終的に終了状態へ向かわない場合は引き返しながら、継続する状態の様々な順序列が一つ一つ徹底的に検索される。検査を必要とするこの状態順列木は成長し、このような検索を行うために必要とされる時間は長くなる。

【0015】非決定性オートマトンにおいては複数の現状態が許容され、入力するシンボルは現状態から後続のいくつかの状態のどれかに変化する結果となる。入力するシンボル列の最後に到着すると、オートマトンの現状態のどれかが終了状態にあるかを決定するための検索が行われる。もしあれば、どのパターンに一致したかを判断することはできないが、入力シンボル列がパターンの少なくとも1つに一致していることが分かる。

【0016】決定性オートマトンは公知のパターンの少量のセットを検索するために用いるには有効であるが、一般的な使用には適さない。従って、元々不正確な検索語に基づいて文書セットから情報を選択的にリトリーブするための効率的で一般的な方法及びシステムへの要望は残り、また、非決定性有限オートマトン技術を非リテラル検索へ応用する機会も残されている。

【0017】

【発明が解決しようとする課題】本発明は上記従来例に鑑みてなされたもので、メトリクス法による、つまり「ファジー」な非決定性有限オートマトンを用いて、非リテラル検索方法を使用することによって、記憶された文書セットに含まれる情報を選択的にリトリーブするための方法及びシステムを提供することを目的とする。

【0018】

【課題を解決するための手段】上記目的を達成するために本発明のシステムは、ユーザー定義による文字列照会を処理装置（102）へ送信するためのデータ入力部（104）を含む。非決定性有限オートマトン（450）はこの文字列照会に対応するように構成されている。記憶された単語（460）はオートマトン（450）に適用され、オートマトン（450）はこのように記憶された単語それぞれに関する不一致値を生成する。表示部（470）はしきい値以下の不一致値を持つ記憶された単語の分類リストをユーザーに提供する。

【0019】本発明の一形態としては、照会と検査中の単語間の差異に基づくオートマトンの状態変化にペナルティー値を加えることにより、システムは不一致値を決定する。このような差異は、余分な文字や、不足文字、近隣の順序が交換した文字や、異なる文字を含む。

【0020】

【発明の実施の形態】以下添付図面を参照して本発明の好適な実施例を詳細に説明する。本発明は、コンピューターに記憶された文書セット又は他のデータセットから、単語又は他の情報をリトリーブすることに関する。ユーザーが検索語に基づいて、文書や文書の一部分をリトリーブしようとする場合がある。しかし、様々な理由によりユーザーが入力した検索語と文書セット中の対応する目的とする文字列とが食い違うこともある。本発明のシステム及び方法によれば、入力された検索語に基づいて、様々な「近似の」文字列を文書セットからリトリーブすることを可能にし、また、リトリーブされた文字列を順番に表示する。このことは、本発明のシステムの順応性の高さを反映している。このリトリーブは非決定性有限オートマトンを使用することによって実現される。

【0021】非リテラル検索システムを設計する問題点の一つは、どのくらい近似していれば文書セット中の単語が検索語と一致していると判断するかを決定することである。本発明では、ある関連問題を解決することによってこの問題を処理する。即ち、単語に特定の変更を加えるために、ルールセットとそれに関連する「ペナルティー」値を設定し、文書中の単語を検索語句と一致するように変更を加えるためにこれらのルールを利用することによって与えられるペナルティーを異なる変換経路についてそれぞれ合計し、その合計値の中で最も小さい値を決定する。そして、この最小ペナルティー値は単語と検索語句間の距離として与えられる。こうして文書セット中の全ての単語を検索語との距離に従ってランク付けすることができる。そして、最小の距離を持つ単語が検出語句に最も近いものと見做され、表示される。本発明のファジー非決定性有限オートマトンは、上記のような最小ペナルティー距離計測をルールセットに従って計算する有効な方法を提供する。

【0022】このルールセットは、多数のルールを含んでもよい。このようなルールは一つのシンボルを他のシ

ンボルで置き換えることを示す「異シンボル」と、シンボルの過不足を示す「過シンボル」及び「不足シンボル」と、単語中の近隣の２つのシンボルの位置順番が交換されてしまったことを示す「逆近隣シンボル」とを含む。例えば、文書セットの単語と検索語句の不一致は、ユーザーが「monkey」という語句を検索したい場合に、誤って「mankey」や更にひどい場合は「makney」とタイプしてしてしまったときに起こる。上記のルールセットは、置き換えや、削除や、追加や、並び変え等のルールの様々な手順を介して、例えば文書セット中の単語である「monkey」を「mankey」や「makney」に変更する手段を提供する。

【００２３】このルールセットでは、多数の種類のルール形態が可能である。従って、文書セットの単語を最終的に同形態に変更するために、様々な他のルール手順がある可能性が高い。例えば「monkey」は、「o」を「a」に置き換える一動作で「mankey」に変更することもできるし、また、「o」を文字削除ルールに従って削除した後に「a」を文字追加ルールに従って追加する二動作で「mankey」にすることもできる。上記の例は全て単一又は近隣の文字操作を利用した例であるが、本発明によれば、複数の文字や他の複雑な動作ルールも含む。

【００２４】ルールが使われる毎に、それに関連するペナルティー値が生じ、その単語を作るためのそれまでの全ての変更の合計ペナルティーに加えられる。それぞれのルールのペナルティー値は、他のルールのペナルティー値と無関係であってもよい。従って、通常は、ある単語から他の単語へ望ましい遷移を実施した場合に、異なるルール手順を使用すれば異なるペナルティーが生じる。上記の例を取ると、「o」を「a」に置き換えるペナルティー５とし、「o」を削除するペナルティーが３で「a」を追加するペナルティーを１とすると、二動作による遷移を実行するペナルティーの方が、一動作により生じるペナルティーよりも少ないことになる。

【００２５】多数のルールが存在し、一つの単語を他の単語へと遷移させるためのこれらルールの応用について多くの手順が考えられるため、その遷移を行うためのルールを使用することによって必ず生じる単一の最小ペナルティー値が存在することになる。また、遷移によるペナルティー値を最小にする様々な同等なルール応用手順が存在することもあれば、その最小ペナルティー値を得るために単一の遷移手順しか存在しないこともある。しかし、どのような与えられたルールセットや、関連するペナルティーや、単語の組みについても最小値が存在する。

【００２６】本発明は、文書セットと、検索語と、ルールセットと、ペナルティーとが与えれている場合に、最小ペナルティー値を決定する技術を提供する。上記の例で言えば、本発明はおそらく「monkey」をリトリーブし、リトリーブされた語句の表示リスト中でその単語を表示することになろう。本発明は、システムが将来の検索における望ましい非リテラル一致を行うための確からしさを向上するために、それぞれのルールの個々のペナルティー値を調整する方法を更に含む。この機能は、過去の検索において直接的又は非直接的にユーザーから得られるフィードバックを利用する。

【００２７】図１において、本発明のシステム１００は、キーボード等の入力装置１０４と、それに接続されたコンピュータープロセッサ１０２（CPU）と、少なくとも一つの第１メモリ１０６と、出力表示装置１１０とを含む。プロセッサ１０２は記憶された文書セットを含む文書記憶媒体１０８にも接続されている。記憶媒体１０８は、タイプや走査によって入力されたり、光学文字読み取り技術を使用して入力されたり、文書記憶に関する当業者によって公知である従来技術を使用して入力された文書セット中に含まれる情報を保存するために使用してもよい。

【００２８】本システムの好ましい例では、文書記憶媒体１０８とプロセッサ１０２は一つのハウジング内にあって、自蔵システムを形成している。変形例としては、記憶媒体１０８は、プロセッサ１０２がネットワークや、ケーブルや、他の情報通信媒体を介してアクセスすることのできる、遠隔地に設置されていてもよい。図２は、本発明のシステム１００の機能を説明するフローチャートである。ステップＳ２００で、プロセッサ１０２は文字列照会（TSQ）を入力装置１０４から受信する。この照会は、文書記憶媒体１０８に記憶された文書セットからユーザーがリクエストする、例えば、検索語又は句、又は他の文字列、又は単一の文字を含む。次にステップＳ２０２で、システム１００は記憶された文書セットのうち、最初の文字列を考慮する。ステップＳ２０２で考慮された最初の文字列は、記憶された文書セットの最初の文字列であっても、本システムの目的のために「最初の」文字列として指定された記憶された文書セット中の文字列の内のどれか一つでもよい。

【００２９】次に非近似値Diをステップ２０６で計算し、そして、ステップＳ２０８でステップＳ２０２で考慮された文書文字列に割り当てる。この非近似値Diを決定するために使用される方法としては、当業者にはいくつかの方法が知られている。従って、本システムにそのような非近似値Diを決定する方法のどれかを用いることができる。

【００３０】好適な例としては、ステップＳ２０６で行われる文字列の非近似値Diの計算は、その値があるしきい値を越えると判断された場合、計算が完了する前に停止する。このしきい値は、例えば、記憶された文書セット中の他の単語について行われた、ステップＳ２０２からＳ２１０のループで得られた値の関数でもよい。計算が中止すると、その文字列はステップＳ２１６で表示されないように、任意の大きなDi値が与えられる。

11

【００３１】好ましい例では、非近似値Diは次の式によって定義される。

$$Di = MIN\sum_{1}^{k} r_k v_k$$

(式1)

ここでr$_k$は、メモリ１０６に記憶されている最初のルールセットのあるルールR$_k$を、文字列照会によって特定されたパターンに一致する文字列を作成するために他のルールの応用を考慮して文書文字列に適用しなければならない回数である。kはルールの数である。w$_k$はそれぞれのルールに割り当てられた正の数のウェイトであり、また、最小化（MIN）は文書文字列を文字列照会のパターンに一致するように変更することのできた、ルールアプリケーション手順の全てについて行われる。

【００３２】第１メモリ１０６に記憶された最初のルールセットは、例えば文字列照会によって特定されたパターンに文書文字列が一致するように、文書文字列を更新するために全体として考慮される操作セットである。典型的な操作としては、余分な文字や、不足文字や、異なる文字、順番が入れ替わった近隣の文字（逆順序の近隣文字）、ケース（格）センシティブな場合の異なるケースの文字、ケースインセンシティブな場合の異なるケースの文字、過不足接頭辞及び接尾辞、又は語基文字等の取り扱いを含む。また、この他に定義された様々なルールを含む別のルールが存在し、使用されることもある。これらの典型的な操作例は、例えば、下記の通り定義される。

【００３３】「余分な文字」とは、文書文字列には含まれるが、文字列照会には含まれない文字のことである。例えば、照会が「misuse」をである場合に文書文字列が「missuse」であるようなことである。この例では、照会中にはない余分な一文字「s」が文書文字列中に含まれている。「不足文字」とは、文字列照会の一部にあるが、文書文字列中に存在しない文字のことである。例えば、照会が「conscious」である場合に文書文字列が「consious」や「concious」である場合である。

【００３４】「異なる文字」とは、文字列照会の望ましい文字の代わりに使用されている、文書文字列の他の文字のことである。例えば、照会が「bounce」である場合に文書文字列が「bounse」や「bownce」等である場合である。これらの例のそれぞれにおいて、一致しないシンボルは一致する文字に置き換えられる。「逆順序の近隣文字」とは、文書文字列中の順番が入れ替わった近隣の文字のことである。例えば、照会が「receive」である場合に、文書文字列が「recieve」であるような場合である。

【００３５】「ケースセンシティブな場合の異なるケースの文字」とは、ケースセンシティブ検索をしている場合、文字列照会のケースと異なる文書文字列中の対応する文字のことである。例えば、文字列照会が「America」である場合に、文書文字列が、初めが大文字の「A」ではない「america」であるような場合である。「ケースインセンシティブな場合の異なるケースの文字」とは、文字のケースが比較的重要でない検索における、文字列照会のケースと異なる文書文字列中に存在する対応する文字のことである。例えば、検索がケースインセンシティブであるときに、文書文字列の全ての文字が大文字で記載されていて、文字列照会が、大文字を含んでも含まなくてもいいような場合である。

【００３６】ケースセンシティブについての別の例としては、単語の全ての文字についてケースセンシティブ／インセンシティブなもの、最初の文字についてのみ（上記例）ケースセンシティブであるもの、そして単語の最初の文字はケースインセンシティブであるがその他の文字は全てケースセンシティブであるもの等がある。ケースセンシティビティーは語幹、語基、接頭辞、及び接尾辞に関する位置に左右される。

【００３７】「過不足接頭辞及び接尾辞、又は語基文字」とは、近隣の文字ブロックが文書文字列へ付加されたり削除されたりすることである。例えば、文字列照会が「exist]である場合、もし「exist」が見つからないとしても「preexist」が見つかる場合があり、これが最も照会に近い単語である。近隣の文字ブロックを構成するこの接頭辞「pre」は、それぞれの文字「p」、「r」、「e」を別々に削除するより、おそらくは軽いウェイト付けで「preexist」から削除すればよい。

【００３８】上記に定義しているそれぞれのルールは、単一のルールであっても、同様の問題を提起するルールのクラスを表すものでもよい。例えば、「異なる文字」ルールは、数字の「１」の代わりに文字「ｌ」である場合にペナルティーを割り当てるルールと、数字「０」の代わりに文字「O」である場合にペナルティーを割り当てるルール等、OCRテキストのためにそれぞれのルール持つルールセットであってもよい。「boune」を得るための余分な文字ルールと、「bounce]を得るための不足文字ルールの応用から「bounce」を「bounse」から得ることができるという意味で、ルールセットR$_k$は特殊用途のためのものである必要はない。

【００３９】本発明のシステムにルールとして含むことのできる他種類の取り扱い及び検索方法は、検索において関連する単語を見つけるシソーラスルールや、検索において発音が同一または殆ど同一である言葉を見つける発音ルールや、言語翻訳ルールや、同じ語基を持つ全ての単語を見つける接頭辞／接尾辞削除ルール等の発見的手法ルールを含んでもよい。またルールは、検索語句のシンボル又はシンボルのストリングが、オプショナルであったり、繰り返されていたり、又はオプショナル及び

繰り返しである等の、特定を含んでもよい。また、ルールは、例えば、「auto-immune」と「autoimmune」が同一又は殆ど同一であるように取り扱われるように、余分な又は削除されたハイフンを取り扱うようにしてもよい。OCR処理はしばしば余分なスペースを挿入したり、インデントされたスペースを削除したりするため、ルールは「ofthe」と「of the」が同一又は殆ど同一であるとの認識をしてもよい。また、ルールは、間違った母音を使用したり、「photo」の綴りを「foto」とするような、一般的な綴りエラーを取り扱うようにしてもよい。他の同業者に公知のルールを本発明のルールセットに加えてもよく、これらは、本発明のシステムのルールセットに含まれるものである。

【００４０】ルールに関連するペナルティーは、例えば入力列の最後の方で現われる文字ほど小さくしてもよく；これは、ファジーオートマトンにおいて、後の段階への変化についてペナルティーを小さくすることにより実施できる。例えば、文字列「tilt」と「tilts」の関係を、「tilt」と「stilt」の関係よりも近似していると考えることである。

【００４１】次のステップＳ２１０では、文書セット中に考慮しなければならない他の文字列が存在するか否かを確認する。図２に示すとおり、文書セット中の文字列のそれぞれに不一致値Diが割り当てられるまで、ステップＳ２０２からステップＳ２１０のループは繰り返される。一例としては、一致テーブルや他のインデックスをシステムが持っていてもよい。その場合、一致テーブルは、全ての文字列に不一致値Diが割り当てられることのないように、文書セット中の選択された文字列を指す。また、別の例ではこの一致テーブルはインデックス検索能力を有し、文書セット中のそれぞれ区別された単語は、テーブル中で単一のエントリーを持ち、それぞれが値Diを割り当てられている。そのため、文書セット中のこれらの単語のかなり多数の例示に直接値を割り当てる必要がない。

【００４２】それぞれのルールに関連するウェイト値w_nは、例えば、最初は予め決められた数でよい。一例としては、全てのルールは当初同じウェイトを与えられ、また他の例ではそれぞれのルールは予測されるウェイトに基づいて与えられる。また、ウェイト値w_nは、ユーザーによって選択されるテキストに一致するようにルールR_nが変換することの確からしさの計測に対して逆関連する。ウェイト値w_nはそれぞれのルールR_nに関連したペナルティー値であることが好ましい。

【００４３】ウェイト値w_nは、ユーザーに対応するものでもアプリケーションに対応するものでもよい。例えば、ある特定のユーザーが検索照会を入力するときよくタイプミスをする場合、このルールに従った最初のウェイトはタイプミスの傾向を反映する。別の例では、OCR技術を利用して得られた文書セットをメモリに記憶する

場合に、OCRが数字の「１」を文字の「l」として、またはその逆として判別することに一般的な不一致が起因するとする。このような一般的なエラーに対するウェイト値w_nを、これらのOCRの状況のための文字置き換えアプリケーションのための過去のデータに基づいて、ある特定の値に予め設定しておくことができる。また、本発明の別の例として、異なるルールが同じウェイト値を持ってもよい。ウェイト値をユーザーによって調整可能、もしくは決定可能にしてもよい。それぞれのルールに関するウェイトは、以下に詳しく示すように、検索結果を表示するうえで重要である。

【００４４】好ましいシステムにおいて、ウェイト値w_nはそれぞれのルールに因るペナルティーである。それぞれのペナルティーは、以下に詳細に説明するように、例えば初期値と、別の１つのルールのウェイト値もしくは複数のルールのウェイト値に基づくレンジと、様式と適応性を示す変化率を司る様々なパラメータに因る属性を持つ。上記のとおり、ペナルティーはそれぞれのルールについて同じ値を設定してもよいし、過去の経験またはそれぞれのルールの相対的な重要性に基づいて設定してもよい。

【００４５】ステップＳ２０８で、一度全ての文書的ストストリングに不一致度Diが割り当てられると、次のステップＳ２１２で文字列セットがリトリーブされる。リトリーブされた文書文字列セットは、文書文字列のそれぞれの例示に関連する位置及び他の情報を含む。システムは、図１の装置１０７等のようなメモリ装置中に記憶された第２のルールセットに基づいて文字列セットをリトリーブする。これらのルールは以下のルールを含む。即ち、不一致値が０である文書文字列のみをリトリーブすること（つまり、完全な一致）、最小の不一致値Diをもつ文書文字列を全てリトリーブすること、最小の不一致値Diをもつ文書文字列の最初のx個をリトリーブすること、不一致値がx未満である文書文字列のみをリトリーブすること、j番目に小さい不一致値Diをもつ文書文字列を全てリトリーブすること、一文書につき最小の不一致値Diをもつ文書文字列を一つリトリーブすること、等である。第２のルールセットは、Di値に影響を与えることもある。適当であれば、そのようなルールは文書全体や文書の特定のセットや一以上の文書の特定の部分における文字列例示に、検索処理の見地及びスタイルに基づいて応用することができる。

【００４６】例えば、それぞれの文書は、使用されると、その文書から得られた文書文字列全てのDi値に影響を与える関連するペナルティーウェイトを持つことができる。文書ウェイトは固定値であっても、ある文書データから決定されるものでも、リトリーブされた文字列からユーザー選択により決定したものであってもよい。文書ウェイトはその文書中の全ての文字列の全ての例示に均一に応用されてもよいし、それぞれ個別の文書文字列

の例示に応じて作られてもよい。例えば、文書に関連するペナルティーウェイトは不一致値がx未満である文書中の文字列例示数から導出してもよい。また、文書に関連するペナルティーウェイトは、文書中の合計テキスト量に対する不一致値がx未満である文書中の文字列例示の割合に応じて導出してもよい。それぞれの文書に１つの関連するペナルティーがある場合は、それぞれの文字列がリトリーブされるために、結果としての文字列の合計ペナルティーが越えてはならない他のしきい値yを設定してもよい。また文書ウェイトは、それぞれの文書に関連してシステムが生成したインデックス又はユーザーが定義したコード中に含まれる情報に基づいていてもよい。

【００４７】好ましい例としては、文字列セットは、それぞれの文書中で起こる個別の文書文字列それぞれについて単一のエントリーを持つ。単文書中で可能な異なる文字列が複数発生することは、与えられた文字列の文書ペナルティーウェイトの計算に寄与するが、リトリーブされた文字列セット中でその文書中で発生する異なる文字列それぞれについて、厳密に一つの要素が確立する。文書中でその文字列が発生する回数と、その文書中でのそのような例示のそれぞれの位置、及び他の情報は、そのセットの一つの要素に関連付けられて保持される。その結果、リトリーブされた文字列セット中で一つの文字列は多数の別個の要素を持つことができる。この場合、ストリングが一回以上発生するそれぞれの文書につき要素は一つである。更に、望ましいしきい値内で文字列照会に「一致」する、異なる綴りを持った、いくつかの別個の文字列が存在してもよく、従って、その文字列はリトリーブされた文字列セットにおいてそれらが発生するそれぞれの文書について別個のエントリーを持つ。

【００４８】一例としては、そのようなペナルティー値、インデックス及びリトリーブされたテキストセットのエントリーは、文書セット又は１つ以上の文書の部分に関連していてもよい。例えば、「文書」の定義は、ページと定義された文書を最終的に含まない他の文書からなる充てられた非周期的なグラフである。他の例では、含まれる最小単位は文字や、文章や、段落等のページ以外のユニットであってもよい。この装置を通して、第２のルールセットは、リトリーブされた文字列セットにおけるどの望ましい細密レベルに基づいても、文書セットに適用することができる。

【００４９】ステップＳ２１４で、リトリーブされた文字列は、リトリーブされた文字列セット中の文字列それぞれに関連する不一致値Diに基づいて並べられる。リトリーブされた最小の不一致値を持つ文字列は、高い不一致値を持つ文字列より優位であることが好ましい。例えば、異なる文書中で起こる場合は、文書文字列から文字列をリトリーブするために用いられる第２のルールセットのルールに基づいて、一つのリトリーブされた文字列

が順番に並び変えられたリストに何回も現われてもよい。また、一つの文書中のストリング発生のためのリトリーブされた文字列セットの全ての要素は、表示目的のために単一のエントリーへ合併される。合併されたエントリーには、そのエントリーが生成されたエントリーの個々の不一致値に基づく不一致値が与えられる。合併されたメトリクス値を得る方法を以下に説明する。表示部はそれぞれの文書についてリトリーブされた文字列全部を、加算された文書の不一致値の順番でソートして表示する。

【００５０】ステップＳ２１６で、表示装置１１０は順番に並べられたリトリーブされた文字列セットを表示する。生成された表示は通常、リトリーブされた文字列を含むページ全体または文書セットのページ部分をリトリーブする前にリトリーブされた文字列をユーザーが見るためのものである。例えば、医療記録の内容で、ユーザーは「Smith」という名前の特定の患者の記憶された医療記録をリトリーブして見ようとしているとする。名前が「Smith」または「Smith」の変形である名前を持つ患者の全ての医療記録をリトリーブする前に、ステップＳ２１６で、検索で存在位置が確認され、リトリーブされた文字列のリストを本発明のシステムは表示する。この表示は、現存する検索システムで一般的に行われているように、「Smith」という名の例示に関する前後関係情報も含む。

【００５１】ステップＳ２１８で表示部１１０から文字列が選択されるのを待つ。この選択は、例えばキーボードや、マウス、またはタッチスクリーン等の入力装置１０４を介してユーザーが行なうことができる。他の例では、選択は自動レポート生成を容易にするためにペナルティー値に基づいて自動的に行ってもよい。この選択は、選択された文字列を有する数ページ又は１ページ又はページ部分を含む文書の部分をリトリーブする他のシステムへ送られる命令となる。本発明のシステムと共に使用することのできるシステムの一例は、米国特許出願番号第０８／０６０，４２９号に開示されている。

【００５２】他の例としては、選択された文字列を処理部へ送り、スペルチェックを行うアプリケーションプログラムでスペルチェックしてもよい。とくに、一度ユーザーが本発明を取り入れたスペルチェックプログラムを起動すると、システムはスペルチェックされた文書中の文字列を見分け、辞書、つまり、文書セットの中を一致するものを求めて検索する。そして、最もよく一致したものをユーザーに向けて以降に説明する要領で表示する。本発明は、外国語ルックアップ、練習、引用、そして辞書システム等を含む様々なシステムで使用することが可能である。

【００５３】本発明の重要な一面は、ウェイト値Wnが本来的に適応的であることである。従って、次のステップＳ２２０はそれぞれのルールおよび表示から選択された

リトリーブされた文字列TS_xに関連するウェイト値W_nを調整する動作を含む。例えば、ユーザーが表示された文字列の一つTS_xを選択したとすると、ステップＳ２０４からＳ２０６で実行されたそれぞれのルールに関連したペナルティーウェイトは減少するか、または調整される。システムが使用され続けると、それぞれのルールR_nに関する選択された文字列のウェイトは、ユーザーおよびアプリケーション環境の両方をシステムが採用する方法で少量上下して調整される。このことは、最終的にユーザーによって選択される近似を首尾よく検出することのできたルールに有効に「報いる」こと、つまり、関連するそれらのペナルティー値を減少させることによって、ウェイト値の調整に基づいてルールを区別するという基本事項を達成する。

【００５４】また、ルールセットの全てのルールR_nに関連するウェイト値W_nは調整される。他の例では、選択された文字列TS_xをリトリーブするために用いられたルールR_nに関連するウェイト値W_nのみを調整する。また他の例では、ルールR_nの選択されたものに関連するウェイト値W_nが調整される。ウェイト値W_nが調整されるある特定

$$C = Dm_{ave} - Di_{ave}$$

ここで、Dm_{ave}は選択されていない文字列mに関連する不一致値の平均であり、Di_{ave}は選択された文字列に関連する不一致値の平均である。上記のとおり、文書セット中のそれぞれの文字列は典型的には関連する不一致値をもつ。図３は摂動法を実施するための好ましいシステムの動作を示すフローチャートである。先ずステップＳ３００で、好ましくは上記に記したアルゴリズムに従って、初期連関測度Cを決定する。典型的には、この最初のステップはリトリーブされた文字列のセットからユーザーが一つ以上の選択を行った後で行われる。次のステップＳ３０２で、一つのルールR_nに関連するウェイトW_nが予め決められた値分だけ減少する。この値は１等の定数又は他の選択された数でよい。減少を行う方法は、例えば減算、除算、又は当業者に知られた他の数学的方法である。この方法を行うためには、他の全てのウェイトは変更されない。調整された連関測度C'はステップＳ３０４で減少されたウェイト値W_n'を用いて決定される。本実施の形態では、このステップＳ３０４における決定は上記で定義されたアルゴリズムに従って行われる。

【００５７】次のステップＳ３０６で、システムはステップＳ３０４で決定された調整された連関測度C'が、ステップＳ３０２での減少に先立ってステップＳ３００で決定された初期連関測度Cよりも上回って増加したかどうかを確認する。もしC'がCよりも増加している場合、ステップＳ３０８でウェイト値W_n'は減少値を保持する。しかし、C'がCを上回って増加していない場合、ウェイト値W_n'はもとの値W_nへ戻り、ステップＳ３１０で予め決められた値だけW_nは増加する。ステップＳ３００からＳ３１０までの処理は、本システムにおいてそれぞれのルールに関連するウェイトについて別々に行われる。増加を行う方法は加法、乗法、又は当業者に知られている他の数学的方法であってもよい。

の方法は本発明の特定例に基づいて調整される。また、システムは、それぞれの後続する検索のためにそれぞれのルールの使用に基づいて調整されたウェイト値が適用されるように、調整されたウェイト値をメモリ中に保持することが好ましい。従って、文書セット中のそれぞれの文字列に不一致値Diを割り当てるステップＳ２０８で、値Diは過去の検索から調整されたウェイト値に基づいて決定される。

【００５５】ウェイト値を調整する目的は、ユーザーが選択したテキストの一致とユーザーが選択しなかったものとを区別して新しいウェイトW_nに到達することである。摂動法（perturbation method）として知られる好ましい方法は、第１のルールセット中の全てのルールのウェイトW_nを調整するために対応測度Cを決定し、使用する。この摂動法では、様々なルールR_nのそれぞれに関連するウェイトは、対応測度Cが増加するように増加又は減少される。本発明を実施することによって決定される対応測度Cは下記の式２で表すことができる。

【００５６】

（式２）

【００５８】また、連関測度CをステップＳ３１０の後に計算し直してもよく、又は改善を確かめるためにCに対してチェックしてもよい。図３に示す処理を行うシステムでは、最初はウェイト値W_nを増加させることを、Cの増加よりむしろCの減少を（又は両方を）チェックすることによって、行うこともできる。摂動法は対応測度を向上するために行われる。ウェイトおよびルールに対する摂動を実行すること含めた本発明の実施において、システムは選択された文字列の不一致値Diと選択されていない文字列間の分離を行おうとする。Cを決定する他の方法を用いることも可能である。

【００５９】システムは、文字列それぞれに関連する選択設定値s_nを含むこともできる。選択設定値s_nは、ある文字列TSが選択されたか否かを表す二値数であってもよい。一例としては、一致がユーザーによって選択された場合は選択設定値s_nに０が割り当てられ、それ以外の場合は１が割り当てる。他の例では、ユーザーはリトリーブされた文字列に優先を順位つけ、近似のものには優位度を示す数値（０又は１に制限されない。）を割り当てる。この場合、ある近似において小さい値ほど興味が大きいことを示す（例えば、１が１番目の選択を示し、２が２番目の選択を示す）。

【００６０】また、他の例では、適切な対応測度Cを以下の方法のどれかによって決定することもできる。即ち、不一致値Diと選択優位度s_n間の相関関係のピアソン積（Pearson product）のモーメント係数、又は距離と

選択優位度間に相関関係がないとする帰無仮説に基づいて計算されたピアソン積のモーメント相関係数の逆の可能性、又は距離と選択優位度間に相関関係がないとする帰無仮説に基づいて計算されたピアソン積のモーメント相関係数の確率を乗じた負の数、又は不一致値Diと選択優位度s_n間のスピアマンの順位相関係数、又は不一致値Diと選択優位度s_n間に相関関係がないとする帰無仮説に基づいて計算されたスピアマンの順位相関係数の逆の可能性である。

【００６１】他の例では、ウェイトW_nが再記憶されてステップＳ３１０で増加されるかステップＳ３０８で減少された後に、ウェイトW_nを正規化してもよい。正規化はウェイトが大きくなり過ぎたり逆に小さくなり過ぎたりするのを防ぐことによって正確に計算を行なえるように、また別々の照会後に決定された不一致値Diが比較可能な値であることを確認するために行われる。

【００６２】正規化処理はウェイト値の全てのセットについて行っても、関連するルールR_nのカテゴリーに対応するウェイト値のサブセットについて行ってもよい。例えば、活字の形状が類似した文字の置き換えに関するウェイトで、他のウェイトとは別の正規化されたウェイトのサブセットを構成することも可能である。正規化処理は、以下の工程の１つ以上の工程を含む。これらの工程とは、即ち、固定平均値又は他の中心傾向測度を得るために、セット中のそれぞれのルールに関連するウェイトW_nにある大きさを加算する工程、固定平均値又は他の中心傾向の測度を得たり又は固定標準偏差又は他のばらつき測度を得るために、セット中のそれぞれのルールに関連するウェイトW_nにある大きさを掛ける工程、又はセット中のあるウェイトを固定値に留めるために、そのセットにおけるそれぞれのルールに関連するウェイトW_nにある大きさを加算する工程、又はセット中のあるウェイトを固定値に留めるために、そのセットにおけるそれぞれのルールに関連するウェイトW_nにある大きさを乗じる工程である。

【００６３】図４は、本発明における非決定性有限オートマトン４５０の状態図を示す図である。オートマトン４５０は、好適な実施の形態において、メモリ、例えばメモリ１０６に記憶されたコンピュータープログラムによって実行され、CPU１０２の動作を制御する。図４で示すこの特定のオートマトン４５０の動作は、ある英単語、例えば「for」をリトリーブするものであるが、他の単語、句、又は他のタイプのシンボルのリトリーブの状態についても手動あるいは自動で構成することができることは言うまでもない。図１の文書記憶媒体１０８に記憶されているような単語４６０のデータベースはオートマトン４５０に適用される。オートマトン４５０で処理された結果は検索リスト表示４７０で用いられる。つまり、記憶された単語４６０のうちどれが、このオートマトン４５０の構成目的である検索語句に最も類似しているかを示す、視覚的又は他の表現による表示を作成する。検索リスト表示４７０は、例えば図１のCPU１０２をプログラムすることにより、従来の方法で実施することができる。

【００６４】図５はオートマトン、例えばオートマトン４５０と、記憶された語４６０のデータベースと、検索リスト表示４７０のための装置による処理を示すフローチャートである。先ずステップＳ５０１で、目的とする検索語又は検索表現のためにオートマトン４５０等のオートマトンを構成する。このようなオートマトン４５０がどのように実行されるかは後に示す。次に、ステップＳ５０２において、文字列を記憶部から得る。そしてステップＳ５０３で、後に詳細を説明するように、オートマトンの状態に予め値を割り当ててオートマトンを初期化する。ストリングの文字をステップＳ５０４で得て、ステップＳ５０５でその文字に応じてオートマトンの状態値が更新される。この動作も後に詳しく説明する。ステップＳ５０６で現文字ストリング中に残された文字があるか否かを決定する。もしあれば、処理はステップＳ５０４に戻り、他の文字を得る。それ以外はステップＳ５０７へ進み、そこでオートマトンの最終状態の値が予め設定されたしきい値未満であるか否かを決定する。この処理も後に詳しく説明する。もししきい値未満であればステップＳ５０８へ進み、現ストリングはヒットとして記憶される。どの場合も、ステップＳ５０９において記憶部中に検査すべきストリングが残されているか否かを決定する。もし残っていれば処理はステップＳ５０２へ戻り、他のストリングを得る。もし検査すべきストリングが残っていなければ、ステップＳ５１０でユーザーのために、ヒットであるストリング全てをリスト表示する。

【００６５】一般的に、有限オートマトンは初期状態と、中間状態と、そして最終状態の３タイプの状態を含む。それぞれの状態は、「on」状態又は「off」状態にセットされ、「初期状態」として処理開始時点で「on」される。一状態から他の状態への遷移は、入力シンボルセットに続くシンボルの同一性に左右される。非決定性オートマトンでは、任意の時点で一つ以上の状態が現状態（つまり「on」に設定された状態）であってもよい。

【００６６】一状態から他の状態への遷移は、ある複数の状態の状況を変化するための環境を特定する遷移ルールセットによって決定される。入力文字がそれぞれ処理される度に、どの状態を「on」としてマークするかを決定するためにそれぞれのルールを検査する。この検査の後、「on」としてマークされた状態のそれぞれがonに設定され、他の状態はoffに設定される。「on」である状態が存在しないときはいつでも、処理において候補に一致するものを認識できなかったと考えられる。

【００６７】このようなオートマトンの状態のセットは、状態の「最終」セット、つまり、もしその状態に到

達した場合は、既に処理された文字がオートマトンの検出目的パターンに一致したことを示す状態として指定される。最終入力信号が処理された後に最終状態が「on」に設定されていることが分かった場合、一連の入力シンボルは最終状態に関するパターンのためのパターンマッチを構成するものと見做される。中間状態とは、初期状態でも最終状態でもないものを言う。

【００６８】オートマトン４５０についてだが、これは非決定性ファジー（又は「拡張」又は「一般化された」）オートマトンであり、「on」又は「off」によってのみ設定される状態を含まない。オートマトン４５０のそれぞれの状態は、その状態を示すメトリクスに関連する。例えば、メトリクスは０（「on-most」状態）から無限（「off-most」状態）までの正の整数である。なお、他のメトリクスシステムを使用してもよい。

【００６９】図４の定義では、現在検索中の文字がある特定の文字、例えば「f」、である場合、ルール４０３等の遷移ルールは、例えば状態４００の第１状態から例えば状態４１０の第２状態への遷移を提供し、そのような遷移によって第２の状態となることによって生じる、例えば「０」等の特定された追加ペナルティーとなる。従って、現文字が「f」であるときの状態４００から状態４１０への遷移は、ルール４０３に基づけばペナルティーの追加がないことになる。ルール４０４の「f」表記は、ルール４０４が「f」以外の文字に適用可能であることを示す。ルール４０２の「ε」表記は新しい文字入力の無い場合にルール４０２が適用可能であることを示す。これらのルールは、（１）第１の入力文字が処理される前に、（２）入力文字が処理された後に（つまり、同様にある文字と次の文字の処理の間に）全てのルールのアプリケーションに従って適用される。ルール４０１の「*」表記は、ルール４０１は状態４００から状態４００へ戻る中間を示すものであるため、ルール４０１がどの文字にも適用可能であることを示す。ルール４０１からルール４０４とルール４１１からルール４１４とルール４２１からルール４２４の斜線の右側の数字は、それぞれ対応する遷移の結果与えられるペナルティーの大きさを示す。このようなペナルティーは、遷移が起こった状態に関するメトリクスを増大させるために加算される。例えばルール４０１は、なんらかの文字が発生場合に、状態４００から状態４００への遷移が状態４００のメトリクスをペナルティー３だけ増加させることを特定する。同様にルール４０４は、「f」以外の文字が発生した場合に、状態４００から状態４１０への遷移が状態４１０のメトリクスをペナルティー４だけ増加させることを特定する。

【００７０】ファジー非決定性有限状態オートマトンの状態は、後に解説するように、ハイフンや曖昧なスペースを処理することによって引き起こされる初期状態へ戻る変化以外は、後の状態から先の状態へ変化することがないように、順番に割り当てられる。オートマトンはステップＳ５０３でいくつかの工程を経て初期化される。先ず、値０がそれぞれの初期状態（つまり、状態４００）に割り当てられる。文字入力なし場合の遷移ルール（つまり、ルール４０２、４１２、４２２）は、順番に値を非初期状態に割り当てるために使用される。この例では、ルール４０２と４１２と４２２は、値５、１０、１５を状態４１０、４２０、４３０へそれぞれ割り当て、一つ以上の最初の文字が入力文字列から抜けている可能性を扱うために使用される。

【００７１】動作において、入力文字を検査する直前および直後に状態が評価される。文字が到着する前の検査は、現在の値と、余分で予想外の文字を含む入力文字列に関する値を加算した値で状態をマークするために行われる。ルール４０１、４１１、４２１、４３１はこのような場合に呼び出され、状態４００、４１０、４２０、４３０それぞれに３を加える結果となる。

【００７２】入力する文字が処理される度に、それぞれの状態がその順番どおりに処理され、その状態について他の状態への遷移のためにどのルールが提供されたかを決定するためにそれぞれのルールを調べ、再マークづけを行ってもよい。現マークよりも低いマークをもつ状態にするルールによって、新しく、より低いマークを与えられた状態となる。例えば、状態４００が値０でマークされ、状態４１０が値５でマークされ、入力文字「f」が処理されるとすると、状態４１０は値０＋０で再マークすることができる。又は、状態４００が値０でマークされ、状態４１０が値５でマークされ、入力文字「g」が処理されるとすると、状態４１０は値４で再マークすることができる。この例のルール４０３、４１３、４２３、４０４、４１４、４２４はこの目的のために使用される。

【００７３】入力する文字が処理されると、可能な状態遷移のそれぞれについて（例えば、状態４１０から状態４２０への遷移）目的の状態（例えば状態４２０）の値を、もとの状態（状態４１０）の値と入力文字列の不足文字によって起こる遷移（ルール４０２、４１２、４２２）のペナルティー値との加算値に等しくにすることによって減少することができるか否かを決定するために、ルールを再検査する。

【００７４】入力文字が処理されると、それぞれの状態に関する値が、最近入力された文字を含むそれまでの最初の入力文字列がオートマトンの目的のパターンの最初の部分によく適合する、エクステント測度となる。処理の途中のある時点で充分に低い値が無い場合は、これ以上の処理においても充分に近似であるものは検出されない可能性が強いため、その入力列に対する処理は中断される。これは、負でないペナルティー値を利用した場合の直接の結果である。

【００７５】枝別れがある場合は、それぞれの枝がそれ

ぞれについて最終状態をもつ可能性がある。複数のパターンが同一であるか否かを同時に確認するように状態機器が作られている場合、様々なパターンのそれぞれに関連する複数の最終状態が存在することもある。また、入力シンボル列全部が処理された後に0値（又は、他の「最近似」値）をもつ最終状態が検出された場合、読み込まれた入力シンボル列は最終状態に関するパターンに完全一致するパターンを構成するものと見做される。それ以外は、最終状態に関する値は、最終状態に関するパターンと入力シンボル列間の不一致度を示し、従って、上記のとおり不一致値Diとして使用される。

【００７６】このように、オートマトン４５０のような、文書中で検索目標となる文字列のためのオートマトンを構成することによって、文書記憶媒体１０８中に記憶される文字列のそれぞれはオートマトン４５０によって処理され、そのような文書文字列それぞれの処理終了時に最終状態に関する値が与えられたしきい値の不一致値未満である場合、そのような文書文字列は「ヒット」と考えられる。従って、このアプリケーションにおいて、オートマトン、例えばオートマトン４５０、は記憶された文字列と目的の文字列間の用語索引の位置を求めるための用途索引システムとして使用される。

【００７７】本発明によると、ファジー非決定性状態装置のそれぞれの状態に関する値は、負の数以外の数に限定される必要はない。例えば、順番に並べられたアーベルセミグループの要素や、順番に並べられたアーベルセミグループの有限外積を検索語と目的のシンボル列間の不一致度を測定するために用いてもよい。従って、数字のメトリクスでなければならない必要はなく、不一致度の階層を構成するために使用することのできる他のメトリクスを同様に使用してもよい。例えば、アーベルセミグループの構成要素の有限列を構成するベクトルを、入力する文字列と目的とするパターンとの相対不一致度を決定するために使用してもよい。

【００７８】そのような場合、ユーザーは目的とするものによって得れらた結果によりよく合致させるために特定の遷移ルールに関連するペナルティーについて感度分析を行うことも可能である。特に、オートマトンのそれぞれの状態に関連する値は、負以外の数であるベクトルから構成され、それぞれの数はルールの異なるペナルティーセットを用いて計算される。異なるペナルティーセットに関する不一致値Diは、ペナルティー値の様々なセットの適切さ加減を評価するために調べられる。ペナルティー値が異なる２組のペナルティーセットがルールの一つのみに割り当てられている場合、類似値の差は特定のルールに対する類似値の感度を示すことになる。このような感度分析は、不一致値が検索語と目的単語間の類似度のユーザーによるランク付けに対応するような方法でペナルティーを計算するために、ペナルティーを順応するように更新するために使用することができる。

【００７９】オートマトンの構成は一致させるパターンを特定するために用いられるユーザーの「規則的な表現」に左右される。規則的な表現は、パターンマッチングやコンパイラー構成等のコンピューター科学のある技術分野で広く使用される。特に、有限状態決定性オートマトンおよび非決定性有限オートマトンは、文字列の処理中にパターン一致を行うために使用される。例えば、参考文献である、Computer Science Press社から１９９２年に発行されているThomas W. Parsons著の「コンパイラー構成入門（Introduction to Compiler Construction）」（第２．６部「規則的表現（Regular Expressions）」、第２．７部「規則的表現と有限状態機器（Regular Expressions and Finite-State Machines）」、他）は、「規則的表現」、有限状態決定性オートマトンおよび有限状態非決定性オートマトンを説明し、定義している。表現は、本目的、つまり（i）0、1、又をそれ以上のシンボル又はシンボル列がそのようは表現が形成するために使用可能であること、（ii）「wild card」を可能な、数字文字、母音、数字大文字等のシンボル又はシンボルのいくつかのある特定のサブセットを表すために用いること、（iii）シンボル又はシンボル列が一回又はそれ以上繰り返して現われる可能性があること、（iv）シンボル又はシンボル列が存在しないこと、（v）シンボル又はシンボル列が存在しないこと、又は一度だけ現われること、又は一回以上繰り返して現われることのために、「規則的表現」であると考えられる。

【００８０】ここで説明するファジーな非決定性有限状態オートマトンは、状態のそれぞれがおそらく0又は1では無い、状態に関連する何かの値を持つという点で、他の非決定性有限状態オートマトンと異なる。状態に関する値は、入力する文字列中の文字が処理される度に変化する。ファジー非決定性有限オートマトンでは、最終状態に関連する値は、処理された入力文字列の文字が規則的表現にどのくらい近く近似するかを示す。

【００８１】なお、ファジー非決定性有限オートマトンを、文字列を検出するためだけに使用されるものではない。例えばコンピュータープログラム中からトークンを検出したり、デジタル血圧計等の計測機器から入力を検出したり、また他の観察機器から情報を検出する等の、他のアプリケーションにおいて、類似のシンボル検査方法を用いてもよい。

【００８２】図４において、有効なオートマトン４５０は、検索表現へ遷移するために現在注目されている単語に行われるべき変化を評価する。そのような変化のそれぞれは、関連する「ペナルティー」又は「ウェイト」を持つ。オートマトン４５０は、目的単語から検索表現への変更を完了するために必要な最小ペナルティーを探す。オートマトン４５０が使用する基本ルールは次のとおりである。（i）目的単語の文字が変更されていない場合、ペナルティーポイントを課さない。（ii）目的単

語の一文字を変更しなければならない場合、４ペナルティーポイントを課する。（iii）目的単語に一文字を加えなければならない場合、５ペナルティーポイントを課する。（iv）目的単語の一文字を削除しなければならない場合、３ペナルティーポイントを課する。上記のルール（i）はルール４０３、４１３、４２３によって実行される。上記のルール（ii）はルール４０４、４１４、４２４によって実行される。また、上記のルール（iii）はルール４０２、４１２、４２２によって実行される。更に、上記のルール（iv）はルール４０１、４１１、４２１、４３１によって実行される。

【００８３】例えば、入力列「are」がオートマトン４５０に適用されたとする。この場合、S_iが検索表現の最初のj個の文字セグメントを、S_0＝「」、S_1＝「f」、S_2＝「fo」、S_3＝「for」のように表し、nがオートマトンの状態の数から１少ない数と等しく、T_iが目的表現の最初のi個の文字セグメントを、T_0＝「」、T_1＝「a」、T_2＝「ar」、T_3＝「are」のように表現し、またmが目的表現全体の長さを表現しているとすると（この場合も３）、オートマトン４５０は「for」と「are」のメトリクス相対距離を決定するために下記の工程を実行する。

【００８４】工程１．ファジーオートマトンを初期化する。０からnまでのそれぞれのjの値について、T_0からS_jまでの距離$D_{0,j}$を計算する（ルール４０２、４１２、４２２を使用）。T_0とS_0は同じ、つまり両方とも空の文字列であるので、T_0からS_0までの距離$D_{0,0}$は０である。次に、検索表現の最初の文字であるS_1に変形するために文字「f」が空の列T_0に加えられるので、T_0からS_1までの距離$D_{0,1}$は５である。同様に、S_2に変形するために２つの文字「fo」が入力列に加えられるので、$D_{0,2}$は１０となり、またS_3に変形するために３つの文字「for」が空の入力列に加えられるので、$D_{0,3}$は１５となる。

【００８５】工程２．入力文字をそれぞれ処理する。１からmまでのそれぞれの値iについて、次のループを実行する。

工程２a．この工程は、第i番目の入力文字が読み込まれる直前に行われる。０からmまでの値jのそれぞれについて、T_jとS_j間の予備の距離$E_{i,j}$を$D_{i,j}$＋３として計算する（ルール４０１、４１１、４２１、４３１を使用）。j＝０の時のこれらの値は、３、８、１３、１８となる。

【００８６】工程２b．この工程は、第i番目の文字が読み込まれる直前に行われる。１からnまでのそれぞれの値jについて、次のループを実行する。

(i)　目的単語の第i番目の文字が検索表現の第j番目の文字と一致した場合、TEMP1の値を$D_{i-1,j-1}$と決定する（ルール４０３、４１３、４２３使用）。その他の場合は、TEMP1の値を$D_{i-1,j-1}$＋４と決定する（ルール４０４、４１４、４２４使用）。

【００８７】(ii)　$D_{i,j}$を$E_{i,j}$の最小値とTEMP1に設定する。

工程２c．$D_{i,0}$＝$E_{i,0}$に設定。

工程２d．１からnの間の値jのそれぞれについて、$D_{i,j}$が$D_{i,j}$＋５より小さい場合、$D_{i,j}$を$D_{i,j-1}$＋５に置き換える（ルール４０２、４１２、４２２使用）。

【００８８】オートマトン４５０のための上記の手順全てを漏れなく行い、入力列が「are」で、$D_{i,j}$、つまりT_iとS_j間の距離をd(T_i,S_j)で簡略に表現し、及び$E_{i,j}$についてe(T_i,S_j)の簡略表現を用いて表すとすると、下記のことが決定される。

決定１．ルール４０２、４１２、４２２を使用し、d(「」,「」)＝０、d(「」,「f」)＝５、d(「」,「fo」)＝１０、d(「」,「for」)＝１５であることを決定する。空の目的列は他の空の列にペナルティー０で遷移することができ、１つの余分な文字を加えることによりペナルティー５で「f」に変位することができる。同様に２つの余分な文字を加えることによる「fo」への遷移はペナルティー１０で行うことができ、３つの余分な文字を加えることによる「for」への遷移はペナルティー１５で行なうことができる。

【００８９】決定２a．（第１の外部ループ：i＝１）ルール４０１を使用し、e(「a」,「」)＝３、e(「a」,「f」)＝８、e(「a」,「fo」)＝１３、e(「a」,「for」)＝１８であることを決定する。目的の列「a」を１つの文字を削除することにより空の列へ遷移することができ（ルール４０１、ペナルティー＝３）、また１つの文字を削除し（ルール４０１、ペナルティー＝３）別の文字を加える（ルール４０２、ペナルティー＝５）ことにより「f」に合計ペナルティー８で変形することができ、また１つの文字を削除し（ルール４０１、ペナルティー＝３）別の２文字を加える（ルール４０２と４１２、それぞれペナルティー＝５）ことにより「fo」に合計ペナルティー１３で変形することができ、また１つの文字を削除し（ルール４０１、ペナルティー＝３）別の３文字を加える（ルール４０２、４１２、４２２、それぞれペナルティー＝５）ことにより「for」に合計ペナルティー１８で変形することができるため、これらの結果は妥当である。

【００９０】決定２b．（第１の外部ループ：i＝１）（第１の内部ループ：j＝１）i＝１、j＝１とすると、上の段落からe(「a」,「f」)＝８であることがすでに分かっている。また、不一致文字を置き換えることにより、「a」を「f」にペナルティー４で遷移することができるため（ルール４０３、ペナルティー＝４）TEMP1＝０＋４＝４であることを決定することができる。$D_{i,j}$を４に設定する（$E_{i,j}$＝８とTEMP１＝４の大きさの最小値）。従って、d(「a」,「f」)＝４。

【００９１】決定２b．（第１の外部ループ：i＝１）（第２の内部ループ：j＝２）i＝１、j＝２とすると、

上の段落から$E_{1,2}$＝１３であることがすでに分かっている。また、「f」を加え（ルール４０２、ペナルティー＝５）、「a」を「o」に変える（ルール４１３、ペナルティー＝５）ことにより、「a」を「fo」に遷移することができるため、TEMP１＝５＋４＝９であることを決定することができる。$D_{1,2}$を９に設定する（$E_{1,2}$＝１３とTEMP１＝９の大きさの最小値）。従って、d(「a」，「fo」)＝９。

【００９２】決定2b.（第１の外部ループ：i＝１）（第３の内部ループ：j＝３）i＝１、j＝３とした場合、$E_{1,3}$＝１８、TEMP１＝d(「」，「fo」)＋４＝１０＋４＝１４を決定する。d(「a」，「fo」)は９であると決定されており、「for」は「fo」に１つの文字を最後に加えるだけなので、この結果は直感的に満足されていることが分かる。

【００９３】決定2c.（第１の外部ループ：i＝１）d(「a」，「」)＝e(「a」，「」)＝３に設定する。

決定2d.（第１の外部ループ：i＝１）d(「a」，「」)＝３、d(「a」，「f」)＝４、d(「a」，「fo」)＝８、d(「a」，「for」)＝９であることは分かっている。この値（３、４、８、９）の順列中では、前者の値から５以上離れている値はないため、変更する必要はない。

【００９４】決定2a.（第２の外部ループ：i＝２）０からmまでのjの値それぞれについて、T_2とS_jの間の予備の値$E_{2,j}$を$D_{2,j}$＋３として計算する（ルール４０１、４１１、４２１、４３１を使用）。これらの値は６、７、１２、１７である。ルール４０１を使用して、e(「ar」，「」)＝d(「a」，「」)＋３＝３＋３＝６を決定する。次に、e(「ar」，「f」)＝d(「a」，「f」)＋３＝４＋３＝７を探す。同様に、e(「ar」，「fo」)＝d(「a」，「fo」)＋３＝９＋３＝１２、また、e(「ar」，「for」)＝d(「a」，「for」)＋３＝１４＋３＝１７を探す。

【００９５】決定2b.（第２の外部ループ：i＝２）（第１の内部ループ：j＝１）i＝２、j＝１とすると、$E_{2,1}$＝７である。また、TEMP１＝d(「a」，「」)＋４＝３＋４＝７である。従って、d(「ar」，「f」)＝７。

決定2b.（第２の外部ループ：i＝２）（第２の内部ループ：j＝２）i＝２、j＝２とすると、$E_{2,2}$＝１２であることが分かっている。また、TEMP１＝d(「a」，「f」)＋４＝４＋４＝８である。従って、d(「ar」，「fo」)は$E_{2,2}$＝１２とTEMP１＝８の大きさの最小値である。従って、d(「ar」，「fo」)＝８である。これは、文字「ar」の「a」を「f」に（ルール４０３、ペナルティー＝４）、「r」を「o」に（ルール４１３、ペナルティー＝４）に置き変えることにより変更できるので、妥当である。

【００９６】決定2b.（第２の外部ループ：i＝２）（第３の内部ループ：j＝３）i＝２、j＝３とすると、$E_{2,3}$＝１７であることが分かっている。「ar」と「for」は最後に同じ文字を持っているので、TEMP１＝d(「a」，「fo」)＋０＝９＋０＝９である。従って、d(「ar」，「for」)は$E_{2,3}$＝１７とTEMP１＝９の大きさの最小値であるので、d(「ar」，「for」)＝９。

【００９７】決定2c.（第２の外部ループ：i＝２）d(「ar」，「」)＝e(「ar」，「」)＝６に設定する。

決定2d.（第２の外部ループ：i＝２）d(「ar」，「」)＝６、d(「ar」，「f」)＝７、d(「ar」，「fo」)＝８、d(「ar」，「for」)＝９であることは分かっている。この値（６、７、８、９）の順列中では、前者の値から５以上離れている値はないため、変更する必要はない。

【００９８】決定2a.（第３の外部ループ：i＝３）０からmまでのjの値それぞれについて、T_3とS_jの間の予備の値$E_{3,j}$を$D_{3,j}$＋３として計算する（ルール４０１、４１１、４２１、４３１を使用）。これらの値は９、１０、１１、１２となる。ルール４０１を使用して、d(「are」，「」)＝９を決定する。次に、e(「are」，「f」)＝d(「ar」，「f」)＋３＝７＋３＝１０を決定する。同様に、e(「are」，「fo」)＝d(「ar」，「fo」)＋３＝８＋３＝１１、また、e(「are」，「for」)＝d(「ar」，「for」)＋３＝９＋３＝１２を決定する。

【００９９】決定2b.（第３の外部ループ：i＝３）（第１の内部ループ：j＝１）i＝３、j＝１とすると、$E_{3,1}$＝１０であることが分かっている。また、TEMP１＝d(「ar」，「」)＋４＝６＋４＝１０である。従って、d(「are」，「f」)＝１０。

決定2b.（第３の外部ループ：i＝３）（第２の内部ループ：j＝２）i＝３、j＝２とすると、$E_{3,2}$＝１１であることが分かっている。また、TEMP１＝d(「ar」，「f」)＋４＝７＋４＝１１である。従って、d(「are」，「fo」)＝１１である。

【０１００】決定2b.（第３の外部ループ：i＝３）（第３の内部ループ：j＝３）i＝３、j＝３とすると、$E_{3,3}$＝１２であることが分かっている。TEMP１＝d(「ar」，「fo」)＋４＝８＋４＝１２であるので、d(「are」，「for」)＝１２。

決定2c.（第３の外部ループ：i＝３）d(「are」，「」)＝e(「are」，「」)＝９に設定する。

【０１０１】決定2d.（第３の外部ループ：i＝３）d(「are」，「」)＝９、d(「are」，「f」)＝１０、d(「are」，「fo」)＝１１、d(「are」，「for」)＝１２であることが分かっている。この値（９、１０、１１、１２）の順列中では、前者の値から５以上離れている値はないため、変更する必要はない。文字列「are」の「a」を「f」に置き換え（ルール４０４、ペナルティー＝４）、「o」を追加し（ルール４１１、ペナルティー＝３）、「r」を保持し（ペナルティー無し）、「e」を削除することによって（ルール４３１、ペナルティー＝５）、「are」から「for」に変形することができるため、上記の結果は妥当である。他の方法では、文字列

「are」の「a」を「f」に置き換え（ルール４０４、ペナルティー＝４）、「r」を「o」に置き換え（ルール４１４、ペナルティー＝４）、「e」を「r」に置き換える（ルール４２４、ペナルティー＝４）ことにより、「for」に変形することができる。

【０１０２】このように、「are」と「for」間の距離は１２である。「are」から「for」へ１２より少ないペナルティーで変形することは不可能である。上記より、一つの単語よりも複雑な表現、つまり「ワイルドカード」、選択的シンボル、繰り返しのシンボル、又は許可されたシンボル又はシンボル列のセットを供給することは、より複雑なオートマトンを必要とする結果となる。そのようなオートマトンは、再結合してもしなくてもよい枝や、ループや、他の遷移方法を含んでいてもよい。また、更新された過不足文字や、入れ替わった近隣の文字や、発音に引きずられた文字の置き換え、光学文字認識エラー（「m」を「rn」と間違える等）、大文字化、文法エラー、接頭接尾語等のファジー特性への対応は、オートマトン中に適切な遷移を含むことにより行うことができる。同様に、文字間や単語間のスペースや、ハイフン、又は光学文字認識装置によって解読できない又は曖昧であると識別された文字に見られるような曖昧さを、対応する遷移ルールを用いて適切に取り扱うことができる。言うまでもなく、文書からの本体テキストだけではなく、文書を要約又は説明するテキストや、文書名、また他のテキストも、上記の方法をもって検索することができる。

【０１０３】一例では、あるシンボルを信頼できるレベルで認識できるOCRシステムを用いた場合、それぞれのシンボルは信頼できる要素及び可能な変形シンボルに関連づけてもよく、ペナルティー値はこの情報から用いることができる。動作の柔軟性を表す他の例を下に記す。〔選択可能文字〕ファジー非決定性有限状態オートマトンは、部分が選択可能であるパターンについて検索を行うことができる。図６は、「for」と「fr」が両方とも完全な一致であると判断されるように、文字「o」が選択可能であるものとして取り扱われる単語「for」を検出するためのファジー非決定性有限状態オートマトンを表す図である。図６は、３つのルール６１２、６１３、６１４を更に含むほかは、図４と同一である。これらの遷移ルールは、状態４００から状態４１０へ変化する代わりに状態４００から状態４２０への遷移を提供するほかは、遷移ルール４０２、４０３、４０４とそれぞれ同じである。これらの遷移ルールの効果は状態４００から状態４１０へ代わることのできる遷移が、状態４１０をバイパスして代わりに状態４２０へ行くことができるようになることである。状態４１０は文字「o」がペナルティー無しで遷移されるもとの状態であるので、新しいルール６１２、６１３、６１４は「o」をペナルティー無しで選択的に削除することをができるようにするものである。

【０１０４】図７は、図６のオートマトンと同じ不一致度測定をもたらす他のファジー非決定性有限状態オートマトンを表す図である。図７は遷移ルール４１２が０ペナルティーを持ち、状態４１０から状態４２０へペナルティー無しで遷移することができること以外は図４と同じである。図７に表すオートマトンは図６のオートマトンより簡略なものであるが、状態４１０から状態４２０への直接遷移は提供されない。

【０１０５】〔繰り返し文字〕ファジー非決定性有限状態オートマトンは、部分の繰り返しが許されるパターンの検索を行うことができる。図８は「for」、「foo r」、「fooor」、「foooor」等が単語「for」と完全に一致しているものと決定されるように、文字「o」を繰り返し可能として取り扱われる単語「for」を検出するためのファジー非決定性有限状態オートマトンを表す。図８は、入力列中の文字「o」が処理されるときにペナルティー無しで状態４２０に留まることを許可する遷移ルール８２１をさらに含む以外は、図４と同じである。

【０１０６】〔選択および繰り返し文字〕ファジー非決定性有限状態オートマトンは、部分が選択及び繰り返し可能なパターンの検索を行う。図９は「fr」、「fo r」、「foor」、「fooor」、「foooor」が全て完全に一致するものとして検出されるように、「o」を選択及び繰り返し可能として取り扱われる単語「for」を検出するためのファジー非決定性有限状態オートマトンを表す。図９は、状態４１０から状態４２０への遷移を０ペナルティーで許可するルール４１２と、文字「o」が処理されているときはオートマトンが状態４２０に留まることを許可する追加ルール９２１以外は、図４と同じである。ルール４２１によれば文字「o」が削除されてもよく、ルール９２１によれば「o」が繰り返されてもよい。

【０１０７】〔文字セット〕ファジー非決定性有限状態オートマトンは、部分が文字のセットから選択されてもよいパターンの検索を行う。図１０は、単語「far」、「fir」、「for」、「fur」を完全に一致しているものとして検出するためのファジー非決定性有限状態オートマトンを表す図である。図１０は、文字「a」、「i」、「o」、又は「u」を受け取ったときに、状態４１０から状態４２０へペナルティー０で遷移することを許可するルール１００１、１００２、１００３を更に含む以外は図４と同じである。これらのルールのそれぞれは同等のペナルティー値、つまりこの場合は０ペナルティーを持つため、ルール４１４についての更なる議論及び追加は不要である。マークされた次の値の最小値化は、文字「i」、「a」、「u」を処理するためにルール１００１、１００２、１００３がルール４１４より優位にあることを裏付けている。

【０１０８】〔OCRエラーに関連するペナルティー〕あ

るOCRが文字「o」を認識間違えることが知られていて、時折、文字「o」を数字の「0」または文字「e」と間違えて認識するとする。「for」、「f0r」、「fer」を完全又は完全に近い一致であるとして認識するオートマトンを作るために上記段落（「文字セット」）で表現した技術を、使用することができる。

【0109】また、文字「o」とOCRシステムが混同するような文字それぞれに対するエラー頻度に基づいて、その文字に関連するペナルティーを定めてもよい。その場合、より一般的に起こるエラーを低いペナルティーに割り当てればよい。例えば、OCRシステムが文字「o」をしばしば「0」と読み間違い、更に殆ど同じ確率で文字「e」を読み間違えるとすると、文字「o」を受信した場合はペナルティー0で、文字「0」を受信した場合は場合はペナルティー1で、「e」を受信した場合はペナルティー2で、第1の状態から第2の状態へ遷移できるように、ファジー有限状態非決定性オートマトンを作ることができる。

【0110】［検索語のセグメントを一致］「psychosocial」と呼ばれる現象についての情報を得るために検索し、「psychosocial」、「psycho」、「social」が完全に一致しているものとして認識されることを期待するとする。ファジー非決定性有限状態オートマトンはこのような検索を行なうことができる。説明の簡略化のために、「psychosocial」を探す代わりに単語「for」についての検索を行うとし、「f」又は「or」のいづれも完全に一致していると認識する。図11はこのような検索を行うファジー非決定性有限状態オートマトンを表す図である。ルール402は入力文字無しで初期状態400から状態410への遷移をペナルティー無しで行うことを許す。このことは、状態410が「or」を完全一致として認識することを許す初期状態であるように振る舞うことを許す。ルール1101は、入力文字が無い場合に、状態410から状態430への変形をペナルティー0で行うことを許す。このことにより、状態410は「f」が完全一致として認識されることを許す最終状態であるように振る舞うことができる。全体の表現「for」が完全な一致として認識されることは明らかであり、従って、望ましい検索を実行することができる。

【0111】同様に、「mother-in-law」の検索を目的とし、「mother-in-law」、「mother-in」、「in-law」、「mother」、「in」、「law」のそれぞれの列が完全に一致しているものとして認識させるとする。このことは、「i」と「l」が検出された状態が初期状態であるように振る舞わせるようにし、初期状態からこれらの状態へ文字を受け取らずに機能するゼロペナルティー遷移を含むことによって、達成することができる。さらに、「r」と「n」の検出によって達成される状態は、これら2つの状態のそれぞれから最終状態へ文字を受け取らずに機能するゼロペナルティー遷移を含むことによって、最終状態のように振る舞うように構成することもできる。

【0112】説明の簡略化のために、「mother-in-law」を検索する代わりに、文字「for」を検出することを目的とするとし、「fo」、「or」、「f」、「o」、「r」をも完全に一致していると認識するとする。図12はこのような検索を行うファジー非決定性有限状態オートマトンを表す図である。ルール402により、入力文字無しで初期状態400から状態410への遷移をペナルティー無しで行うことができる。これにより、状態410は「o」が最初の文字であるような役割を果たすことを許す初期状態であるように振る舞うことができる。またルール1201により、入力文字無しで状態400から状態420への遷移をペナルティー無しで行うことができる。このことにより、状態420が、「r」が最初の文字であるような役割を果たすことを許す初期状態であるかのように振る舞うことを可能にする。ルール1202は、入力文字無しで、状態410から状態430への遷移をペナルティー無しで行うことを許す。このことにより、状態410は最終状態であるように振る舞うことができる。ルール422は、入力文字無しで、状態420から状態430への遷移をペナルティー無しで行うことを許す。このことにより、状態420は最終状態であるように振る舞うことができる。

【0113】図13は図12に示すファジーオートマトンが解決するものと同様の問題を解決し、さらに、「fr」を完全に一致しているものとして認識する。図11の場合とは異なり、完全一致は一連の近隣の文字で構成されている必要はない。図13に示すオートマトンにおいて、状態400は初期状態であるが、ルール402と412は状態410と420も初期状態であるように振る舞うことを可能にする。同様に、ルール412と422は、本来は状態430のみが本当の最終状態であるところ、状態410と420も最終状態であるように振る舞うことを可能にする。

【0114】［逆順序の近隣文字］図4に示すファジーオートマトンは、文字「for」を完全一致するものとして認識するためのものである。目的単語が「ofr」（近隣の文字「f」と「o」の順序が入れ替わった例）の場合や、目的単語が「fro」（近隣の文字「o」と「r」の順序が入れ替わった例）の場合、不一致値は8（過不足文字に対するペナルティーの合計）となる。近隣文字の順序の入れ替わりは、タイプエラーやスペルエラーとしてしばしば起こるものであるため、このようなエラーには比較的低めのペナルティーレベルを割り当てることが望ましい。

【0115】図14は「for」を完全一致（つまり、ペナルティー0）として認識し、「f」と「o」又は「o」と「r」のどちらかが入れ替わった場合、ペナルティー6を割り当てるオートマトンを表す図である。なお、

「fo」が入力列の最初に見つかった場合、状態４００、４１０、そして４２０は連続して値０が割り当てられる。しかし、「of」が入力列の最初に見つかった場合、状態４４０は値０が割り当てられ、状態４２０で値６が割り当てられる。

【０１１６】同様に、一連の状態４１０、４２０、４３０は文字列「or」をペナルティー無しで処理するために用意されている。しかし、一連の状態４１０、４５０、４３０は文字列「ro」をペナルティー６で処理するために用意されている。過不足文字のウェイト合計よりも小さいウェイトを持つ別のルートを提供する処理は、逆順序の近隣文字に対して特別のペナルティーを割り当てるために、ファジーオートマトンにおける連続する経路のいかなる組み合わせに対しても用いることができる。

【０１１７】〔ハイフン又は曖昧なスペースを含む目的単語〕OCRエラーは単語間のスペースを検出する際にしばしば起こる。例えば、単語「for」は、「f」と「or」の間にスペースを含む２つの単語として読まれるかも知れない。しかし、「f」と「or」の間に故意にスペースが挿入されているか否かを知る方法はないため、「for」、「f」、「or」のいづれかを検索しているときにヒットとするすることが望ましい。

【０１１８】別の例を考えてみると、例えば、キー入力された文書中で「self-」が行の最後にあり、「confident」が次の行の最初に見つかったとする。コンピューターがそのハイフンは単語「self-confident」の必要部分なのか、単語が次の行へ続くためにキー入力されたものなのかを判断できないとする。この問題は、目的テキスト中のハイフンを曖昧なスペースとして取り扱うことによって解決される。文書中の単語「self-confident」は、検索中に「selfconfident」、「self-confident」、「self」、又は「confident」のいづれとも完全に一致するものと結論される。

【０１１９】図１５は目的単語「for」を認識するオートマトンであるが、これはが文書中に「for-lorn」や、「be-for」や、「for loan」（曖昧なスペースを含む）や、「be for」（曖昧なスペースを含む）や、「f-or」や、「f or」（曖昧なスペースを含む）が見つかった場合に、これらを完全な一致であるとして結論付ける。図１５に表すオートマトンを適用する前に、ハイフンが目的単語の最初に挿入され、もう一つのハイフンが目的単語の最後につけられる。これらの付加された前及び後ハイフンを含むそれぞれのハイフン又は曖昧なスペースが見つけられると、ハイフン又は曖昧スペースが見つかる度に行われる遷移ルール４３３（以下に説明する）及び変更が行われる。

【０１２０】図１５は、最終状態として機能する状態４６０を含む。ルール４３３は、状態４３０から状態４６０へのペナルティー０の遷移が入力文字を処理することなく行われることを示す。ルール１５０１は、ハイフン又は曖昧スペースが処理された場合に初期状態４００からそれ自身へのペナルティー０の遷移のためのものである。ルール１５０１に関連する記号「－／ρ」は、遷移ルール１５０１が使用できるように、ハイフン又は曖昧スペースが入力列中で見つかる度に状態４００に関する値を０にリセットすることを示す。ルール１３０１、１３０２、１３０３は、ハイフン又は曖昧スペースが入力列中に見つかったときに初期状態４００へ戻す遷移をする。ここでまた、記号「－／ρ」は、ハイフン又は曖昧スペースが見つかった結果としてルール１３０１、１３０２、１３０３のどれかが使用された場合に、状態４００に関する値を０にリセットすることを示す。ルール１５０１、１３０１、１３０２、１３０３の効果は、目的単語のハイフン又は曖昧スペースに続く文字を、入力列の最初の文字として認識されるように取り扱うことができるようにすることである。

【０１２１】ルール４３３に関連する「－／０」は、入力列中にハイフン又は曖昧スペースが見つかった結果としてルール４３３を適用する場合に、ペナルティー０が使用されることを示す。このルールの効果は、目的単語中のハイフン又は曖昧スペースがすぐに後に来る文字を、入力文字列の最終文字であるかのように取り扱うことである。余分なハイフンを目的単語の最初と最後に挿入する（上記のとおり）ことにより、オートマトンが初期化され、目的単語中の全ての元の文字が処理された後にルール４３３が適用されることを確認することができる。目的単語が完全に処理された後の状態４６０に関する値は、ハイフン又は曖昧スペースを処理した後の状態４３０に関する最小値となる。

【０１２２】なお、ルール１３０１、１３０２、１３０３はオートマトン中でループを形成する。以前は、既に考慮されたループはある一つの状態からそれ自身への遷移を提供するものであった。ハイフン又は曖昧スペースを処理することにより引き起こされる初期状態へ戻る遷移は、状態４００、４１０、４２０、４３０、４６０に順番を割り当てる場合に、考慮にいれなくてもよい。

【０１２３】また、ここまでの説明は単一の正則表現（例えば「for」）の場合についてなされたが、複数の正則表現（例えば「for love」）の場合は、特に、複合検索表現における構成正則表現への一致語句を含む文書それぞれのための全メトリクス値の計算及び表現に関して、特別の技術が必要である。複数の通常のファジー検索表現は、１つ以上のファジーな検索語句により構成される。検索表現中のファジー検索語句それぞれは、いくつかの可能な検出された単語に関連している。これらの検出された単語は、ファジー有限状態非決定性オートマトン技術によってファジー検索語句に「近い」と認識される、目的文書セット中の複数の文字列である。検出された単語のそれぞれは、処理によって個別のメトリクス値が与えられる。このメトリクスは検出された単語とそ

れに関連するファジー検索語句との間の距離の測度である。既に説明したとおり、この検出された単語のメトリクスは、それぞれがペナルティー値を有するルールのセットからの遷移ルールの最小アプリケーションの関数として発生するペナルティーの累計として定義される。

【０１２４】検索表現中のファジー検索語句それぞれについて、ユーザーは認識処理において有効である０またはそれ以上の検出された単語と、それらを含む文書表現を選択する。他の例では、ファジー検索語句に最も近似するN個の検出単語等、適切な検出された単語を自動的に選択するためにユーザーが確立した主義に基づいたアルゴリズムを用いている。このように検出された語句の少なくとも一つは、ファジー検索表現中の全てのファジー検索語句に基づいて検出された全単語セットの中から選ばれなければならない。他の例では、このように検出された単語の少なくとも一つは、ファジー検索表現中の複数のファジー検索語句それぞれに基づいて検出された、特定の単語セットそれぞれの中から選ばれなければならない。ファジー検索語句間の望ましい関係を成立させる、OR, AND及び近似拘束等の論理及び他の演算子が要求された選択セットに影響を与えてもよい。

【０１２５】一つ以上の選択された検出単語を含む文書を、ユーザーのためにリスト表示する。それぞれの文書は、文書中に含まれるそれぞれの検出単語の例示カウントを有する。表示の前に文書セットをソートすることを可能にしたり、複合検索表現に対してリストに表示された文書それぞれの適合性をユーザーが評価する助けとなるように、文書毎にメトリクス値が割り当てられる。そしてユーザーは、文書を見たり、印刷したり、更なる検索をしたり、又は他の方法で使用するために、リスト中から文書を選択することができる。

【０１２６】好適な例としては、文書に関連するメトリクスは単位間隔［０．０，１．０］間の数字として計算される。この文書メトリクスの計算は、構成検出単語の個別のメトリクス値と例示カウントに基づいて行われる。ここで構成検出単語とは、複合検索表現中で「近似」している決定し、適性に基づいてユーザーによって選択され、文書に含まれると言うことが分かったものである。具体的には、計算は６つのステップを含む。第１ステップではそれぞれの検出単語のペナルティー値を下記の式によって異なる形式に遷移する。

【０１２７】

new_metric＝（１．０／１．０＋（ペナルティー／１２．０））　（式３）

従って、ファジー検索語句と検出単語の完全一致に対応するペナルティー値０は一致のnew_metoric値となる。不完全一致に対応する０ではないペナルティーはnew_metoric値が０に近いことを示す。ペナルティー０を一致値へ、より大きなペナルティーを０に近い値へ遷移する処理は全て、このステップで行うのが適切である。定数１２．０は、上記の例で提示された提案ペナルティー値を使用している場合に、new_metoric値に妥当なレンジを提供するものであるが、他の定数を用いることも可能である。

【０１２８】第２ステップでは、文書中の適切な検出単語セットに関連するnew_metoric値のセットから、new_metoric値の最大値を文書それぞれについて記録する。この最大値は文書それぞれについて決定されるので、ファジー検出語句に最も近似する、文書中の適切な検出単語のnew_metoric値を記録することになる。別の例では、この最大値は複合検索表現中のそれぞれのファジー検出語句のnew_metoric値の最大値の積であってもよい。この場合、ある文書中で０ではない例示カウントを持つ適切な検出単語のみが考慮され、その文書中の適切な検出単語で表現することができないファジー検索語句のためにデフォルトの低new_metoricタイプ値が使用される。この値は、正規化後に文書上でメトリクス値を制限するために使用される。このことは、第６ステップに関連して後で説明する。

【０１２９】第３ステップでは、検出単語のnew_metoric値とそれぞれの例示カウントとを組み合わせる。それぞれの文書について、文書中のそれぞれの検出単語の例示カウントとnew_metoric値の積の合計が計算される。別の例では、例示カウントはそれぞれ１づつ増加し、それらに関連するnew_metoric値で掛ける前にそれらの対数をとってもよい。

【０１３０】第４ステップでは、考慮対象である全ての文書セットの合計の最大値を決定する。第５ステップでは、それぞれの合計を、第４ステップで決定した最大値で割る。こうすることにより合計が正規化され、これら正規化された全ての値が単位間隔中に存在するようになる。

【０１３１】第６ステップつまり最終ステップでは、全ての正規化された合計を最大の正規化された合計（この合計は、文書に統一した値でなければならない）を持つ文書の最大new_metoric値で掛ける。上記説明はテキスト文書を検索するためのものである。ここで開示されるファジー有限状態非決定性オートマトンは、例えば数学検索木等の他の状況下で最良の一致を探すことに応用することもできる。例えば、ファジー有限状態非決定性オートマトンを一般化された最適化問題における深さ優先木探索に応用することができる。

【０１３２】また、図２を参照して説明したウェイトを変更してもよく、例えばルール４０１等の遷移ルールに関連するペナルティー値に変えてもよい。

【０１３３】

【他の実施形態】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピューター（また

はCPUやMPU）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM等を用いることができる。

【０１３４】また、コンピューターが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピューター上で稼働しているOS等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１３５】さらに、記憶媒体から読出されたプログラムコードが、コンピューターに挿入された機能拡張ボードやコンピューターに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１３６】図１６は本発明にかかるプログラムコードが格納された記憶媒体のメモリマップの一例を示す図で、各モジュールに付記した符号は図２及び図５のステップ番号に対応する。なお、上記の説明は英単語を例に挙げて説明したが、日本語単語における過不足文字、同音異議語、「づ」と「ず」等の音声による文字の置き換え、漢字の間違い、入力間違い等にも本発明が適用可能であることはいうまでもない。

【０１３７】上記の説明は好適な実施の形態の動作を表現するものであり、本発明の範囲を限定するものではない。関係分野の当業者には、発明の精神と範囲を越えない程度で、上記の説明から多数の変形例が可能であることが明らかであろう。

【０１３８】

【発明の効果】以上述べた様に本発明によれば、メトリクス法による、ファジーな非決定性有限オートマトンを用いて、非リテラル検索方法を使用することによって、記憶された文書セットに含まれる情報を選択的にリトリーブすることができるという効果がある。

【図面の簡単な説明】

【図１】本発明における実施の形態にかかるシステムのブロック図である。

【図２】本発明におけるシステムの動作を示すフローチャートである。

【図３】本発明に応用可能な計測方法の動作例を示すフローチャートである。

【図４】本発明における非決定性有限オートマトンの状態図である。

【図５】本発明におけるオートマトンを使用した処理のフローチャートである。

【図６】本発明におけるオプショナル文字の処理を行うことを特徴とする非決定性有限オートマトンの状態図である。

【図７】本発明におけるオプショナル文字の処理を行うことを特徴とする別の非決定性有限オートマトンの状態図である。

【図８】本発明における繰り返し文字の処理を行うこと特徴とする非決定性有限オートマトンの状態図である。

【図９】本発明における繰り返し及びオプショナル文字の処理を行うことを特徴とする非決定性有限オートマトンの状態図である。

【図１０】本発明における重複文字の処理を行うことを特徴とする非決定性有限オートマトンの状態図である。

【図１１】本発明における重複文字列の処理を行うことを特徴とする非決定性有限オートマトンの状態図である。

【図１２】本発明における重複文字列の処理を行うことを特徴とする別の非決定性有限オートマトンの状態図である。

【図１３】本発明における重複文字列の処理を行うことを特徴とする第３の非決定性有限オートマトンの状態図である。

【図１４】本発明における近隣交換文字列の処理を行うことを特徴とする非決定性有限オートマトンの状態図である。

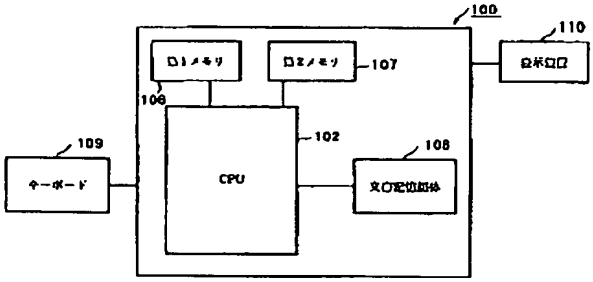
【図１５】本発明におけるハイフン及び曖昧なスペースの処理を行うことを特徴とする別の非決定性有限オートマトンの状態図である。

【図１６】本発明にかかるプログラムコードが格納された記憶媒体のメモリマップの一例を示す図である。

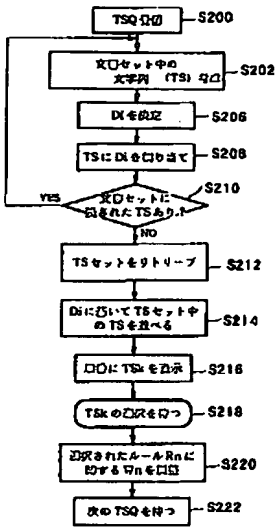
【符号の説明】

１０２　CPU
１０４　キーボード
１０６　第１メモリ
１０８　文書記憶媒体
１１０　表示装置

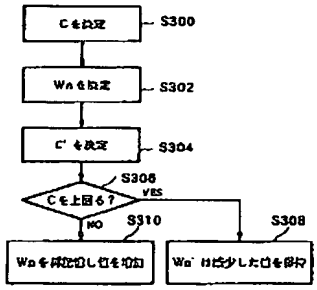
【図1】



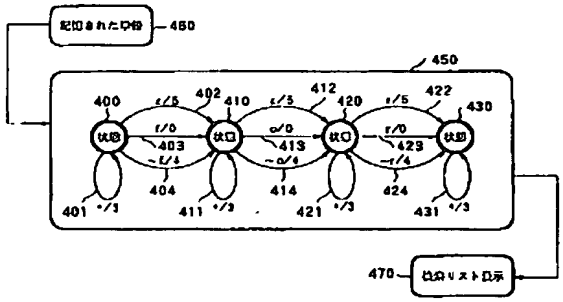
【図2】



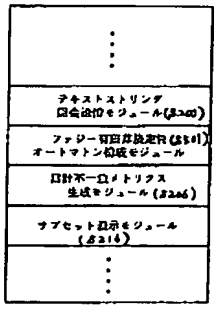
【図3】



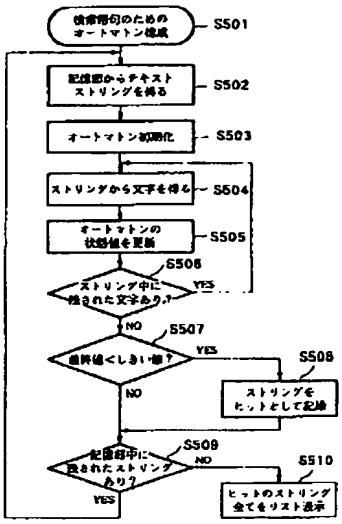
【図4】



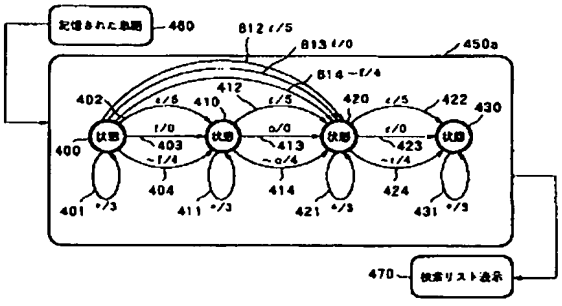
【図16】



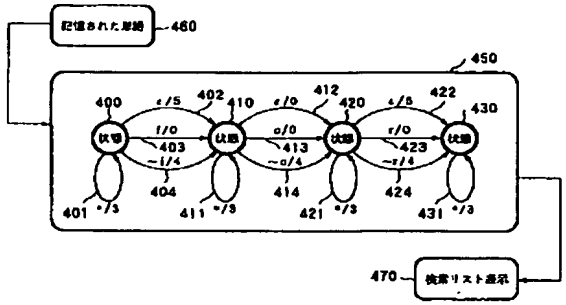
【図5】


検索語句のための
オートマトン構成
S501
記憶部からテキスト
ストリングを得る
S502
オートマトン初期化
S503
ストリングから文字を得る
S504
オートマトンの
状態値を更新
S505
S506
ストリング中に
残された文字あり?
YES
NO
S507
YES
S508
ストリングを
ヒットとして記憶
NO
S509
記憶部中に
残されたストリング
あり?
NO
S510
ヒットのストリング
全てをリスト表示
YES

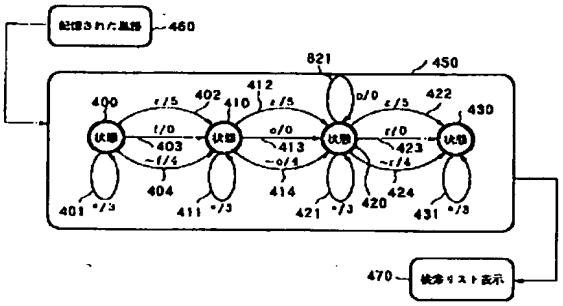
【図6】


記憶された単語
460
812 c/5
813 t/0
450a
814 ~t/4
412
402
c/5
410
c/5
420
c/5
422
430
状態
400
t/0
403
~t/4
o/0
413
~o/4
t/0
423
~t/4
404
414
424
401
*/3
411
*/3
421
*/3
431
*/3
470
検索リスト表示

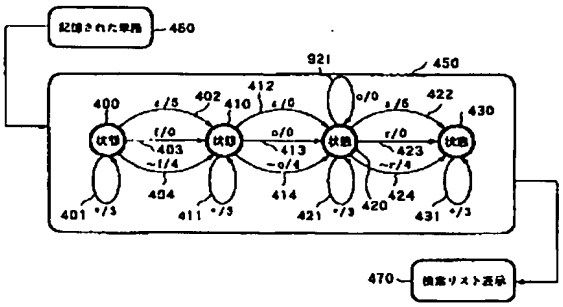
【図7】


460
450
400
402
410
412
420
422
430
403
404
413
414
423
424
401
411
421
431
470

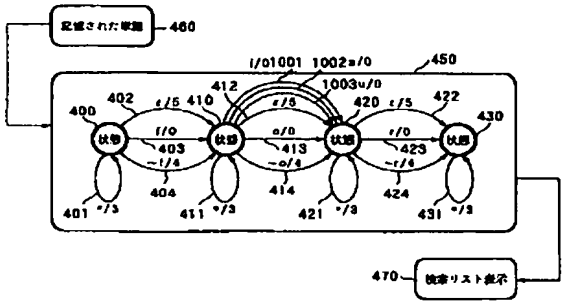
【図8】


460
821
450
400
402
410
412
420
422
430
403
404
413
414
423
424
401
411
421
431
470

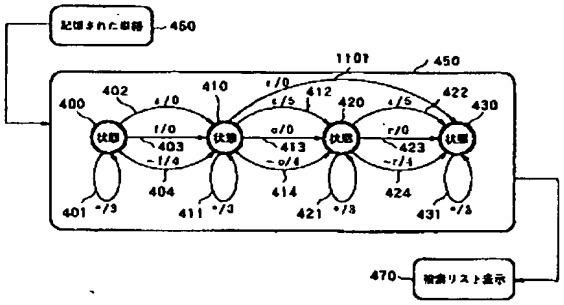
【図9】


460
921
450
400
402
410
412
420
422
430
403
404
413
414
423
424
401
411
421
431
470

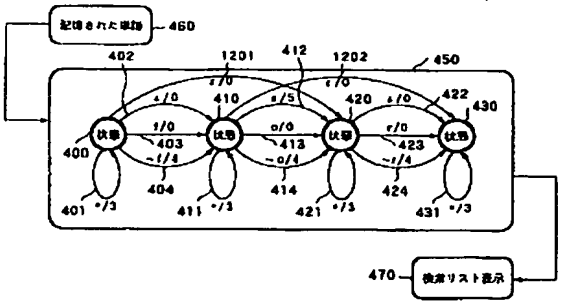
放泯隧亜


460
450
402
412
410
420
422
400
430
403
413
423
404
414
424
401
411
421
431
470

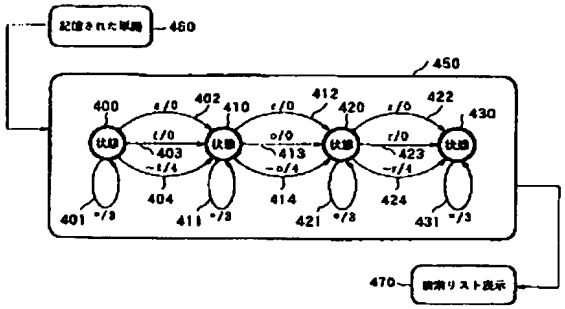
【図１１】


460
1101
450
402
410
412
420
422
400
430
403
413
423
404
414
424
401
411
421
431
470

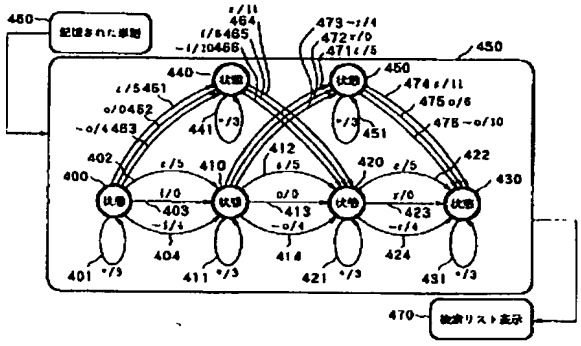
【図１２】


460
402
1201
412
1202
450
410
420
422
400
430
403
413
423
404
414
424
401
411
421
431
470

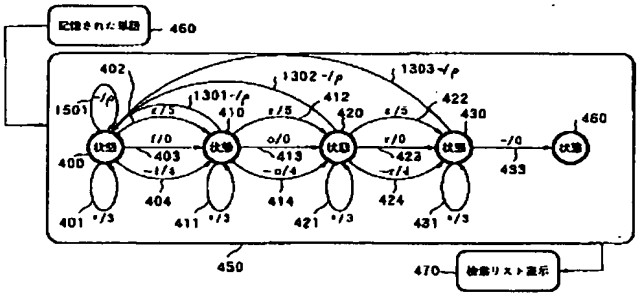
【図１３】



【図１４】



【図１５】



49 50

フロントページの続き

(72)発明者 ハリー・ティ・ガーランド
アメリカ合衆国 カリフォルニア州
94022、 ロス アルトス、 ピュリシマ
ロード 27555